(1)

CONFIGURATION COMPARISION ANALYSIS FOR THE AFIT/AAMRL ANTHROPOMORPHIC ROBOTIC MANIPULATOR

THESIS

Steven L. Parker
Captain, USAF

AFIT/GA/AA/88D-08

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GA/AA/88D-08

CONFIGURATION COMPARISION ANALYSIS FOR THE AFIT/AAMRL ANTHROPOMORPHIC ROBOTIC MANIPULATOR

THESIS

Steven L. Parker
Captain, USAF

AFIT/GA/AA/88D-08

AFIT/GA/AA/88D-08

CONFIGURATION COMPARISION ANALYSIS FOR THE AFIT/AAMRL ANTHROPOMORPHIC ROBOTIC MANIPULATOR

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Astronautical Engineering

Steven L. Parker, B.S.A.E.
Captain, USAF

December, 1988

Acknowledgments

I would like to thank all the professors at AFIT whose never ending patience helped me to understand a small part of how things work in the physical world. In particular, I would like to extend thanks to Dr. Spenny and Dr. Leahy for working so hard to put together a meaningful robotics program. The robotics courses are not only interesting, but also the work you do in the classroom is used for further research. This creates a feeling of pride among the students knowing that their work is contributing to the field, and not just a wasted student exercise.

It wouldn't be right for me not to thank my wife Sonya for her behind the scenes effort to maintain the sanity in our house. Working fulltime, raising two children, and maintaining a household by yourself, was not a task I would burden you with willingly.

Steven L. Parker

Table of Contents

List of Figures

List of Tables

Abstract

A method of calculating mechanical efficiency was developed as a means of comparing the performance of different types of manipulators. As an initial approach to this problem takes into account inertial and gravitational terms of the robot configurations in addition to a variable payload. The method included developing a numerical integration algorithm to calculate the work done by each manipulator at any point in that manipulator's workspace. The efficiencies of two robotic manipulator configurations that are candidates for the design of the AFIT, AAMRL, Anthropomorphic Robotic Manipulator, ($A^3_{\cup}RM$), were analyzed. The two designs were a serial open link direct drive manipulator, and the closed parallel kinematic chain direct drive manipulator design by Dr. Asada at M. I. T. The difference between the mainpulators was actual mass and kinematic design.

The efficiency measure used to analyze both manipulators was based on the magnitude of the total work done by the manipulator to move a payload a prescribed distance. The effects of a variable mass payload on efficiency have now been individually examined for the cases when the arm has been "tuned" for some nominal payload by means of compensating for gravity, making the robotic configuration invariant, and decoupling the manipulator's dynamic equations of motion.

An algorithm was developed for calculating the mechanical efficiency for different robotic mainipulator configurations. When the manipulators are gravity compensated for a nominal payload, their efficiency increases dramatically, even when the payload is varied from nominal. In addition, when the configuration is tuned for dynamically decoupling and configuration invariance, efficiency is improved. Finally, for most of the reachable workspace of the manipulators, the parallel manipulator is the most efficient.

CONFIGURATION COMPARISION ANALYSIS FOR THE AFIT/AAMRL ANTHROPOMORPHIC ROBOTIC MANIPULATOR

I. Introduction

A method was needed to analyze the performance of different robotic manipulator configurations. This method will be used to select the configuration used for the AFIT/AAMRL anthropomorphic robotic manipulator (A^3RM), which will be used for, (one application), research in human telepresence. The goal of the (A^3RM) is a manipulator system capable of achieving human levels of reach, speed, and load carrying capacity. An analysis of two mechanical configurations for the design of a general two degree of freedom robot was conducted in order to determine which configuration would be better suited for the A^3RM system. As a "first-look" analysis, the foundation was set to expand this research into more complex systems.

Through careful selection of the physical configuration and mass distribution characteristics of a robotic mainpulator, greater efficiency and minimal control complexity can be achieved. Methods of achieving minimal control complexity has been addressed in the literature [1]. These methods have generally been applied to manipulators with constant or zero payloads. For the general manipulator designed to be flexible enough to handle multiple operations, constant payload assumptions are not suitable. Therefore, the effects of variable payloads were considered in the analyisis.

In addition, development of analytical tools to measure the performance of these types of manipulators has not been presented to any great extent. Shin-Min

Song and Jong-Kil Lee in their paper *The Mechanical Efficiency And Kinematics Of Pantograph Type Manipulators*, [14], examine using mechanical work as a measure of efficiency. Song and Lee's work looked at massless systems in a limited area of the workspace. Their research did not present a total representation of the efficiency of the manipulator configurations considered. First, Song and Lee's conclusion that the parallel was 100% efficient, is true only for a poriton of the workspace. Second, inertial terms and the mass of the manipulators in the gravitational terms, i.e. a massless arm was assumed, in the dynamic equations of motion were not considered.

1.1 Background

The comparison of different type of robotic manipulator configurations required a somewhat generic tool that could be equally applied to all types of configurations considered. The tool applied was the determination of mechanical efficiency in the workspace of a robot configuration. This efficiency was determined for a serial and a parallel kinematic configuration. The effects of "tuning" on the efficiency were also considered. These configurations are well known in robotics and are defined in the following sections.

1.1.1 Serial Manipulator The serial manipulator is an open kinematic chain with torque producing actuators on each revolute joint, as shown in Figure 1.1.

These robots have one joint and one corresponding link for every degree of freedom. This is the most widely used configuration for robotic applications. Further discussion on these type of robots can be found in [1], [3], and [13].

1.1.2 Parallel Manipulator The parallel manipulator configuration, as shown in Figure 1.2, is a closed link kinematic chain robot having at least one redundant degree of freedom in its structure.

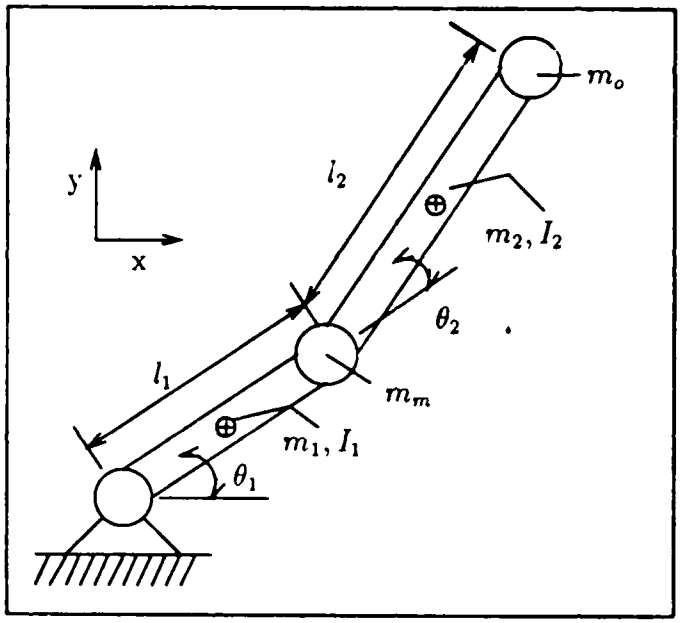

Figure 1.1. Serial Manipulator Configuration

Furhter discussion on these configurations can be found in [1] and [13].

1.1.3 *Tuning* Control of a manipulators position, velocity, and acceleration is a very complex topic. The equations of motion that model the movement of the robot are non-linear, coupled, ordinary differential equations. In practice, classical control of this motion is often inadequate, [1]. A large amount of current research is directed toward using modern optimal control techniques to compensate for these inadequacies. However, with careful design analysis these non-linearities and coupling effects can be reduced to a minimum. Reducing these non-linear effects is called "tuning". Complete elimination of non-linear and coupling effects allows the system to be modelled as a single input - single output system for each actuator. This type of system would be the least complicated and the easiest to control. The analysis technique for reducing nonlinear and coupling terms in the equations of motion was developed by Haruhiko Asada and Kamal Youcef-Toumi in their book *Direct-Drive Robots: Theory and Practice* [1]. Asada and

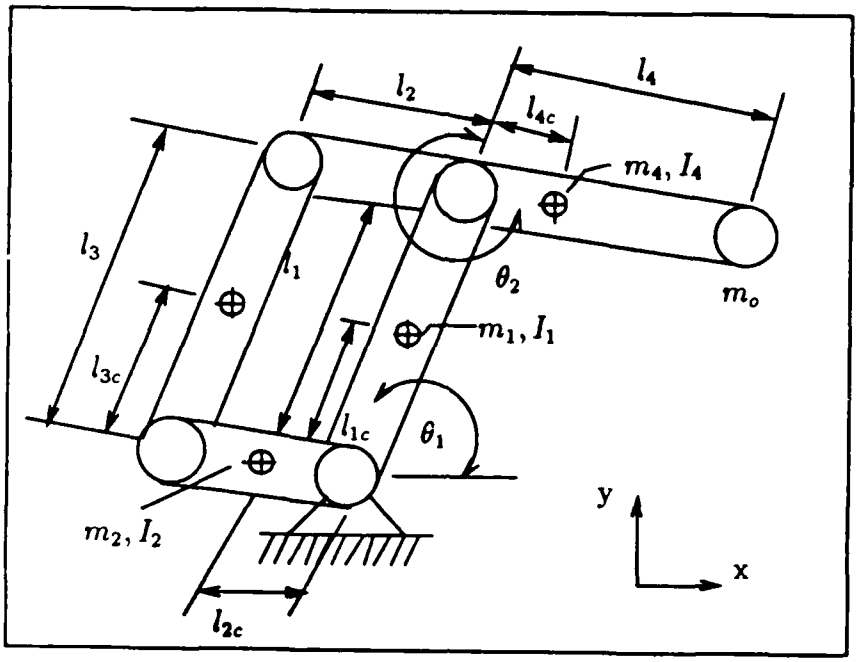

Figure 1.2. Parallel Configuration

Youcef-Toumi were able to design a manipulator with no dependence on joint angle, defined as "configuration invariance", and no coupling effects, defined as "dynamically decoupled". This resulted in equations of motion that were linear, uncoupled ordinary second order differential equations with constant coefficients for the nominal payload for which the manipulator was tuned. When the payload varies from the nominal case, all the non-linear terms of the equations of motion reappear. The dynamics of the robotic system are no longer linear and may require model based control techniques, [8], to obtain adequate feedback control.

1.1.4 *Gravity Compensation.* In addition to dynamic decoupling and configuration invariance, the gravitational terms of the equations of motion can be eliminated for some nominal load [13]. The "gravity compensation" constraints are compatible with the dynamic decoupling and configuration invariance constraints and can also be included in the term "tuning". Therefore all the constraints can be applied to the design of the manipulator simultaneously.

1.1.5 Definition of Mechanical Efficiency. According to Beer and Johnston, [2, page 411], mechanical efficiency is defined as the amount of work out of a system divided by the the amonut of work put into the system. Work is defined in the plane as,

$$W = \int_1^2 \vec{F} \cdot d\vec{r} \tag{1.1}$$

where $\vec{F}$ is a force acting through a displacement $d\vec{r}$. When a moment is acting on a rigid body the work done can be written as,

$$W = \int_1^2 \vec{M} \cdot \vec{\omega} dt \tag{1.2}$$

where $\vec{M}$ is the moment and $d\theta$ is the angular displacement in radians.

In Song and Lee's analysis, [14], the output work was found from moving the payload parallel to the gravity vector some finite distance. Because the force involved was constant the work was easily integrable to be $mg\Delta y$, where Δy is the displacement.

1.2 Method Of Approach

Using Song and Lee's mechanical work as a measure of efficiency, the serial and parallel robotic arm configurations were evaluated. The equations of motion for these well-known manipulators were readily available, [13], making these configurations well suited for this analysis. Inertia and gravitational terms of the equations of motion were included in the analysis.Structural, dimensional, and motor requirements for the arm design was not a part of this analysis.

A basic design had to be drawn up to set a baseline for the configurations. The manipulator is being designed to emulate human arm motion. The dimensions of the configurations were assumed to be similiar to those of the 50th percentile Air Force male [4]. Once the basic design was determined, link mass was redistibuted in order to construct manipulator configurations that were gravity balanced, con-

figuration invariant. Efficiencies of each configuration were then evaluated for the following cases:

- Only gravity terms of the equations of motion.
  - No arm mass included.
  - Arm mass included.
- Inertia terms only.
  - No configuration invariance tuning applied.
  - Configuration invariance tuning applied.
- All terms of the equations of motion included.
  - Nominal case - no tuning.
  - Gravity compensated case.
  - Gravity compensated and configuration invariant case.

Redistribution of mass to decrease the effects of non-linear terms in the equations of motion is often refered to as "tuning". The tuning is dependant on a particular payload. The cases studied here assumed that the manipulator was tuned for zero payload. The effect of non-zero payload on efficiency was then determined. Efficiency was found in the first quadrant of the reachable workspace after determining that the other quadrants were symmetrically similiar to quadrant one. To include inertia terms, the trajectory was assumed to be in a straight line at constant acceleration.

1.3 Summary

In every case, the parallel manipulator was more efficient than the serial. When gravity balancing was imposed, efficiency of both the serial and parallel configu-

rations increased. Also, configuration invariance tuning made the parallel robot more efficient.

This document is organized in the following manner. In chapter II, the human arm characteristics are defined along with how these characteristics are applied to the robot arms and how the tuned cases are achieved. Then, chapter III presents the analysis of the mechanical efficiency. Finally, chapter IV considers the conclusion and recommendations. In the appendices A and B, the actual computer programs developed are included.

II. Anthropomorphic Manipulator Design

The choice to design a manipulator to emulate the motion of a human arm was made to be compatible with future research efforts at the Air Force Institute of Technology (AFIT) and Armstrong Aerospace Medical Research Lab (AAMRL). One application for using an anthropomorphic robotic manipulator is as a surrogate in a telepresence application. Making the robot arm anthropomorphic makes the telepresence task more intiutive to the human operator. In this application the human operator is removed from a hostile/hazardous environment and operates the robot by strapping on a sensored exoskeleton. Movement of the the exoskeleton results in the robot executing the same movement.

Specifically, the design will emulate the motion of a 50th percentile Air Force male which was documented by H. T. E. Hertzberg at AAMRL [4]. The intended robot design will have three degrees of freedom, all revolute, including two orthogonal degrees of freedom at the shoulder and one degree of freedom at the elbow. The elbow axis of rotation and one shoulder axis of rotation are parallel. Using these two rotations as independent coordinates, any reachable position in a plane can be described by these coordinates. With these two degrees of freedom, any plane in the robot's work volume can be represented, alieviating the neccessity of having to look at the three degree of freedom equations of motion for this first-look analysis. When gravitational terms are omitted from the equations of motion, a plane perpendicular to gravity is simulated. When gravity is included, a plane parallel to gravity is simulated.

This chapter will address the design requirements for human arm emulation specifically in terms of dimension, range of motion, speeds, and accelerations. From these specifications, the dimensions, weights, kinematics, and dynamic equations of motion for both a serial and parallel two degree of freedom anthropomorphic configuration are characterized.

2.1 *Human-like Specifications*

To perform as a human surrogate, the manipulator must be capable of reproducing the movements and load carrying capacity of an average human arm. And the manipulator must as a minimum be able to reach what the average arm can. The intent of this manipulator design is to emulate the motion of a 50th percentile Air Force male. These specifications, except for velocity and acceleration, were developed by H. T. E. Hertzberg at the Armstrong Aerospace Medical Research Laboratory [4]. The dimensions of this average male arm are summarized in Table 2.1 [4, page 499].

Arm Section	Length (inches)
Shoulder to Elbow	14.3
Elbow to Handtip	18.9
Wrist to Handtip	7.5

Table 2.1. Average Human Arm Dimensions, [4]

The range of motion specifications are in Table 2.2 [4, page 545].

Motion	Range (degrees)
Shoulder Bend	249
Shoulder Twist	182
Elbow Bend	142

Table 2.2. Average Human Range of Motion, [4]

These specifications of the the range of motion for the average human arm define the human workspace.The average human arm velocities and accelerations are found in Eugene I. Rivin's book *Mechanical Design Of Robots* [13, page 10]. The experimental data for these specifications is found in B. A. Petrov's work [12], but unfortunately this has never been translated from Russian to English. Human arm speed and acceleration are summarized in Table 2.3.

Motion	Maximum angular speed (rad/s)	Maximum angular acceleration (rad/s^2)
Shoulder Bend	7.0	70
Shoulder Twist	10.0	120
Elbow Bend	17.0	300

Table 2.3. Average Human Arm Speed and Acceleration, [13,12]

The motions described in these Tables 2.2, and 2.3 refer to shoulder bend, shoulder twist, and elbow bend. These motions are defined as follows:

Shoulder Bend The angle the shoulder joint moves through in a plane parallel to the body sagital plane.

Shoulder Twist The angle through which the shoulder joint rotates about a vertical axis, i.e., across the chest.

Elbow Bend The angular motion of the elbow joint. Not the twisting motion of the forearm.

2.2 Configurations

From the human arm specifications, the basic design of the serial and parallel configuration can be determined. Inherent to each configuration is a link that goes from the shoulder to the elbow, which will be called length l_1, and a forearm, which will be called length l_2 for the serial arm, as labeled in Figure 1.1, and length l_4 for the parallel as labeled in Figure 1.2.

To compare the configurations, it was necessary to postulate a design for each configuration of manipulator to fix parameters of length, mass, centers of gravity, and moments of inertia. For the sake of simplicity, the cross sections of the manipulators, as shown in Figure 2.1,were assumed to be rectangular and hollow, and are the same size for all links. This cross section is based on human arm dimensions from [4].

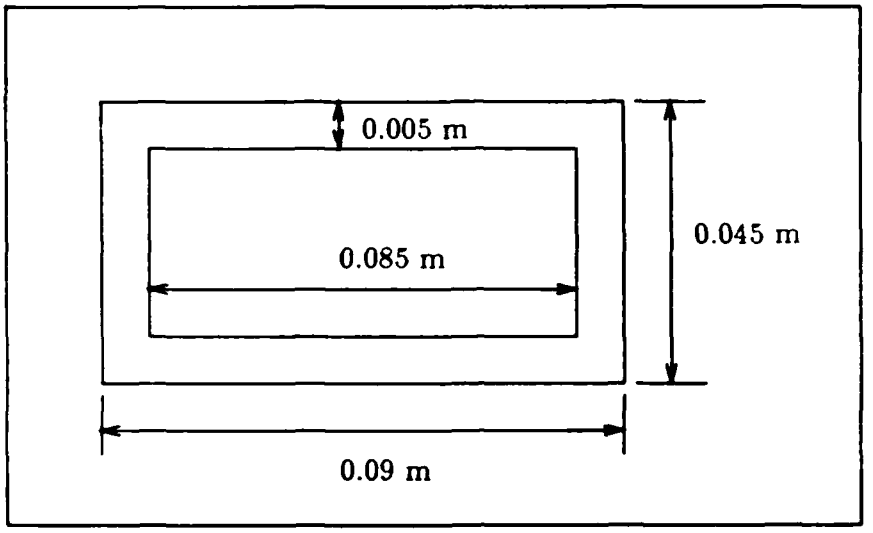

Figure 2.1. Rectangular Cross Section of Each Link

The material used to construct the links was assumed to be readily obtainable aluminum with density $\approx 2.8 \times 10^3 \frac{kg}{m^3}$. An aluminum with this density could reasonably be selected as an inexpensive material to manufacture the arm links. From this density and volume of the links, the mass of each link is obtained, i.e., $mass = \rho \times cross\ sectional\ area \times length.$

Based on the human arm dimensions, Table 2.1, link 1 is then set equal to 0.36322 meters, and link 2 is 0.6705 meters in length. From these lengths the mass of link 1 was then calculated to be equal to 0.6667 kg, and link 2's mass equals 1.3333 kg, twice that of link 1.

All motors were assumed to be of 6 kg. A review of current motor specifications showed that the Moog model 304-008 brushless D. C. motor was capable of achieving near human arm sppeds, see Table 2.3 of more than 10 radians/sec. The motor is probably slightly undersized for the task at joint one, but is within practical approximation limits.

As a reasonable estimate of the payload, it was assumed that the mass of the payload was equal to 1/5th of the serial arm mass, where the serial arm mass was made up of the mass of link 1 and 2, and the mass of the motor at joint 2, as shown in Figure 1.1. The resulting payload was 1.6 kg. Structural integrity was not at issue for the development of the basis for evaluation of mechanical efficiencies of manipulator configurations.

In order to present a more comprehensive comparison, the variables chosen to be independent generalized coordinates will be the joint angles of the serial manipulator, i.e. θ_1, and θ_2, see Figure 1.1. These angles are consistently used for both configurations.

From this average human arm data, the weights, and dimensions of each configuration can be determined to within a reasonable proximity. Design specifics can now be discussed.

2.2.1 Serial Configuration. Most currently available robotic manipulators have serial open loop chain kinematics. Serial manipulators are such that each link is attached to the end of the previous link allowing for:

- More reach
- Movement in another degree of freedom
- More positioning flexibility, multiple paths to same cartesian position. (i.e. redundant degrees of freedom).

The motors or actuators for serial manipulators can be located at the joint of the the link driving the link as in the case of "direct-drive" robots, remotely located through some kind of transmission mechanism such as gears and shafts. Direct-drive robots are particulary well-suited for high speed applications. Human arm emulation requires high joint speeds. Therefore, the A^3RM prototypes' drive sys-

tems were assumed to with direct-drive motors. For the serial configuration the motor driving link 2 was assumed to be located at joint 2.

The following notation is used to describe the parameters of the serial manipulator:

m_o: The mass of the payload. (Nominally $m_o = 1.6\ kg$)

m_1: The mass of link 1. ($= 0.6667\ kg$)

l_1: The length of link 1. ($= 0.36322\ meters$)

m_2: The mass of link 2. ($= 1.3333\ kg$)

l_2: The length of link 2. ($= 0.6705\ meters$)

m_m: The mass of the motor at joint 2. ($= 6.0\ kg$)

This notation will be used throughout this text unless otherwise indicated. The centers of gravity of each link will additionally subscripted with the letter c.

2.2.2 Parallel Configuration. The parallel manipulator configuration arises from using a mechanical linkage to transmit the torque of a motor physically located off the joint axis of the joint of a particular link. In addition, the configuration is called parallel because the linkage is constructed such that it connects to form a parallelogram. A parallelogram is chosen to simplify the relationship of the angles of the links. This method of transmission places the motor weight (which generally is quite significant) closer to the base of the robot arm, relieving the arm of the additional torque caused by the motion of the motor.

In this analysis, the transmission linkage is used to locate the motor driving joint 2 at joint 1. Two links are added to the serial configuration to construct the parallel. In order to maintain the reach of the average human arm, the length of the link extending from the end of link 1 to the payload, is unchanged. This link has been designated as link 4, but is indeed equivalent in length to link 2 of the

serial manipulator. The transmission mechanism then makes up the parallelogram. The side extending back from the elbow is assumed to be 1/4th of the length of link 4, which equals the length of link 2. This number was chosen as a reasonable compromise between the values used in the Asada parallel mechanism ,[1], and the Kazerooni parallel direct drive robot, [6]. The other additional link, link 3, is naturally the same length as link 1 to complete the parallelogram.

The notation used for the serial manipulator is extended to the parallel manipulator configuration. When necessary to distinguish between the two, an addtional subscript p will be used for the parallel case and an s for the serial.

2.3 *Forward Kinematics*

The forward kinematic equations are used to relate a position in a base coordinate frame to the rotating and translating body axis coordinate of the payload or end effector of the robot arm. For the configurations tested, two rotational joints in a plane are considered. The kinematic equations are equivalent for each configuration.

The kinematic equations are as follows:

$$x = l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) \tag{2.1}$$

$$y = l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) \tag{2.2}$$

Knowing the joint angles of the robot arm the position of the end effector is expressed in the based coordinate frame coordinates (x,y). These equations describe the points of the x-y plane that are reachable by the manipulator. This area of the plane is commonly called the robot's "workspace".

2.4 *Inverse Kinematics*

Typically, the position of the end effector is known in base coordinates and the joint angles are desired. The kinematics equations that solve for the joint

angles given the base coordinate values are called the inverse kinematic equations. For a two dimensional problem this is easily done in closed form, [13, page 42], [3, chapter 3]. For this problem the equations are given in Rivin's book, [13, page 44], as follows:

$$\theta_2 = cos^{-1}\left(\frac{(x^2+y^2)-(l_1^2+l_2^2)}{2l_1l_2}\right) \tag{2.3}$$

$$\theta_1 = tan^{-1}\left(\frac{-(l_2 sin\theta_2)x + (l_1 + l_2 cos\theta_2)y}{(l_2 sin\theta_2)y + (l_1 + l_2 cos\theta_2)x}\right) \tag{2.4}$$

Because the inverse tangent function is not well behaved for 360 degrees of revolution, it is more convenient to use a relationship that looks at the signs of both x and y. In the Fortran programming language this function is known as ATAN2. It requires an input of two arguments, x and y. If the point in question is in quadrants 1 and 4 of the x-y plane, then Equation 2.4 is sufficient. For quadrant 3, θ_1 equals the result of Equation 2.4 minus 180 degrees. In quadrant 2, 180 degrees must be added to the result of Equation2.4 to find θ_1. Using these relationships the joint angles are obtainable from any portion of the manipulators workspace.

2.5 Dynamic Fquations of Motion.

The instantaneous torques required for each actuator at a point in the work space are found through rigid body dynamics. These relationships are the equations of motion representing the manipulators. Robot arm dynamic equations depend on the joint angles, the velocity, and the acceleration of those angles. In general, ignoring drive system dynamics, these equations are of the form:

$$\vec{\tau} = \mathbf{D}\ddot{\vec{\theta}} + \vec{\mathbf{h}} + \vec{\mathbf{g}} \tag{2.5}$$

where $\mathbf{D}$ is the inertial tensor, $\vec{h}$ is a vector of coriolis and centrifugal terms, and $\vec{g}$ is a vector of gravitational terms.

A MACSYMA, [15] program was developed to generate the symbolic equations of motion for both serial manipulators and closed link kinematic chain ma-

nipulators. The code and its development, implementation, and limitations are included in appendix A.

The equations for each configuration follow

2.5.1 *Serial Configuration.* Based on terms (lengths, masses, and moments of inerita) defined in Figure 1.1, the equations of motion of the serial manipulator are as follows:

$$\begin{aligned}
\tau_1 &= [m_1 l_{1c}^2 + I_1 + (m_2 + m_m) l_1^2 + m_2 l_{2c}^2 + I_2 \\
&\quad + m_0 l_1^2 + m_o l_2^2 + 2 l_1 (m_2 l_{2c} + m_o l_2) cos\theta_2]\ddot{\theta}_1 \\
&\quad + [m_2 l_{2c}^2 + I_2 + m_o l_2^2 + l_1 (m_2 l_{2c} + m_o l_2) cos\theta_2]\ddot{\theta}_2 \\
&\quad - 2 l_1 (m_2 l_{2c} + m_o l_2)\dot{\theta}_1 \dot{\theta}_2 sin\theta_2 - l_1 (m_2 l_{2c} + m_o l_2)\dot{\theta}_2^2 sin\theta_2 \\
&\quad + (m_1 l_{1c} + m_2 l_1 + m_o l_1)\, g\, cos\theta_1 + (m_2 l_{2c} + m_o l_2)\, g\, cos(\theta_1 + \theta_2) \qquad (2.6) \\
\tau_2 &= [m_2 l_{2c}^2 + I_2 + m_o l_2^2 + l_1 (m_2 l_{2c} + m_o l_2)\, cos\theta_2]\ddot{\theta}_1 \\
&\quad + (m_2 l_{2c}^2 + I_2 + m_o l_2^2)\ddot{\theta}_2 + l_1 (m_2 l_{2c} + m_o l_2)\dot{\theta}_1^2\, sin\theta_2 \\
&\quad + (m_2 l_{2c} + m_o l_2)\, g\, cos(\theta_1 + \theta_2) \qquad (2.7)
\end{aligned}$$

2.5.2 *Parallel Configuration.* Based on terms (lengths, masses, and moments of inerita) defined in Figure 1.2, the equations of motion of the parallel manipulator are as follows: (The generalized coordinates used in these equations are the same common independant coordinates used in the serial manipulator.)

$$\begin{aligned}
\tau_1 &= [m_1 l_{1pc}^2 + I_1 + m_3 l_{3c}^2 + I_3 + m_4 l_1^2 + m_o l_1^2 \\
&\quad - (m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4)\, cos\theta_2]\ddot{\theta}_1 \\
&\quad - [(m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4)\, cos\theta_2]\ddot{\theta}_2 \\
&\quad + (m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4)\, sin\theta_2 \dot{\theta}_2^2 \\
&\quad + (m_1 l_{1c} + m_3 l_{3c} + m_4 l_1 + m_o l_1)\, g\, cos\theta_1 \qquad (2.8) \\
\tau_2 &= [m_2 l_{2pc}^2 + I_2 + m_3 l_2^2 + m_4 l_{4c}^2 + I_4 + m_o l_4^2 + I_o \\
&\quad - (m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4)\, cos\theta_2]\ddot{\theta}_1
\end{aligned}$$

$$+[m_2 l_{2pc}^2 + I_2 + m_3 l_{2p}^2 + m_4 l_{4c}^2 + I_4 + m_o l_4^2 + I_o]\ddot{\theta}_2$$
$$-(m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4)\ sin\theta_2 \dot{\theta}_1^2$$
$$-(m_2 l_{2c} + m_3 l_2 - m_4 l_{4c} - m_o l_4)\ g\ cos(\theta_1 + \theta_2) \qquad (2.9)$$

2.6 *Achievement of Minimum Control Complexity*

As stated earlier, the analysis of these manipulators included enhancing the efficiency by minimizing control complexity. The efficiency enhancement techniques investigated were:

- Dynamic Decoupling
- Configuration Invariance
- Gravity Compensation

These techniques are compatible, i.e. they can be implemented simultaneously with the exception that the serial configuration can't be dynamically decoupled. However, unless the configuration of the manipulator can change dynamically with payload, these conditions can only be met for some nominal payload. This research effort is the first time that the effects of a varying payload, other than the nominal "tuned" payload, have been examined. This section will address the theory behind minimum control complexity, and how it was applied to the serial and parallel manipulators.

2.6.1 Design for Dynamic Decoupling and Configuration Invariance Asada and Youcef-Toumi [1, chapter 4] presented a method for reducing the control complexity of a manipulator by simplifying the dynamic equations of motion. These equations are in general highly coupled, non-linear, second order ordinary differential equations. Asada and Youcef-Toumi have examined how to design a manipulator's mechanical configuration to achieve decoupled dynamics and configuration invariance. The dynamic decoupling eliminates the off-diagonal terms of the inertia

tensor and simultaneously eliminates the coriolis and centrifugal terms resulting in a single input - single output control algorithm for each actuator. The configuration invariance reduces the non-linear equations to linear, ordinary differential equations with constant coefficients.

Dynamic Decoupling. Ignoring the motor dynamics, the equations of motion for a manipulator with revolute joints can be expressed as [1]:

$$\tau_i = D_{ii}\ddot{\theta}_i + \sum_{j \neq i} D_{ij}\ddot{\theta}_j + \sum_j \sum_k \left(\frac{\partial D_{ij}}{\partial \theta_k} - \frac{1}{2}\frac{\partial D_{jk}}{\partial \theta_i} \right) \dot{\theta}_j \dot{\theta}_k + \tau_{gi} \qquad (2.10)$$

where D_{ij} is the i-j element of the inertia matrix and τ_{gi} is the gravity term. For dynamic decoupling the off diagonal elements of the inertia matrix are set to zero by careful configuration design. These constraints are different for each configuration and are presented in full detail after further deveopment. With the off diagonal terms equal to zero, the equations of motion reduce to:

$$\tau_i = D_{ii}\ddot{\theta}_i + \sum_k \left(\frac{\partial D_{ii}}{\partial \theta_k}\dot{\theta}_i \dot{\theta}_k - \frac{1}{2}\frac{\partial D_{kk}}{\partial \theta_i}\dot{\theta}_k^2 \right) + \tau_{gi} \qquad (2.11)$$

Eliminating the off diagonal inertia terms has significantly reduced the complexity of the equations of motion. However, there are still non-linear terms arising due to the dependence of the manipulator on joint orientation.

Configuration Invariance. Another simplification that can be done to the dynamic equations of motion is to make the manipulator independent of spatial orientation. This simplification eliminates the non-linear terms of the equations of motion, Equation 2.10, except for the gravity terms. This implies from Equation 2.10 that the following relationship must be true:

$$\left(\frac{\partial D_{ij}}{\partial \theta_k} - \frac{1}{2}\frac{\partial D_{jk}}{\partial \theta_i} \right) = 0 \qquad (2.12)$$

Substituting Equation 2.12 into Equation 2.10 the equations of motion become:

$$\tau_i = D_{ii}\ddot{\theta}_i + \sum_{j \neq i} D_{ij}\ddot{\theta}_j + \tau_{gi} \qquad (2.13)$$

Except for the gravity terms, these equations are linear with constant coefficients. Combining the configuration invariance with the decoupling completes the simplification process yielding linear (except for gravity terms), decoupled equations of motion.

2.6.2 *Design for Gravity Compensation* In addition the reduction of nonliearity obtained from configuration invariance and dynamic decoupling, the nonlinear gravitational terms in the equations of motion can be completely eliminated. The basic idea is to balance the manipulator about each actuated pivotal point. This is usually done by using the weight of the actuator as a counterbalance to the weight of the link (or set of links) it is driving. In order to do this counterbalancing, the link has to be extended beyond its joint of rotation. This may physically impede the motion of the manipulator, limiting its workspace unless carefully designed to avoid this problem.

2.6.3 *Applied Minimal Control* In order to conduct a fair comparison of both manipulators, parameters of length and mass must established. The following conditions were used to establish a baseline configuration:

- The human arm dimensions were imposed on each manipulator.
- For the parallel manipulator, the added length, l_2, was arbitrarily chosen to be $l_2 = l_4/2$ for all cases of the analysis.
- Any added linkage that was required to implement tuning was added at a constant mass/unit length.
- The weight of the motors was used for balancing when applicable.
- The nominal payload used for tuning was $m_o = 0$.

2.6.4 *Serial Manipulator.* The tuning conditions were applied to the serial manipulator. Looking at the equations of motion, Equation 2.7, it can be deter-

mined that only configuration invariance and gravity compensation are achieveable. The off-diagonal terms of the inertia tensor are all always positive and hence cannot be eliminated. Therefore, two tuning conditions have to be met (from Equation 2.7).

For configuration invariance the following condition must hold:

$$m_2 l_{2c} + m_o l_2 = 0 \tag{2.14}$$

This is also the condition for link 2 to be gravity compensated. For the nominal baseline, however, the payload m_o is zero. Therefore, the condition for configuration invariance reduces to:

$$l_{2c} = 0 \tag{2.15}$$

For gravity compensation the condition is:

$$m_1 l_{1c} + m_2 l_1 + m_o l_1 = 0 \tag{2.16}$$

With the all the lengths and masses fixed in the baseline and with $m_o = 0$, the only variable then is l_{1c}. Solving for l_{1c} the condition becomes:

$$l_{1c} = \frac{-(m_2 l_1)}{m_1} \tag{2.17}$$

Note that without the configuration invariance condition imposed, gravity compensation is not achievable for the serial manipulator.

To enforce the configuration invariance and gravity compensation, the locations of the motors were moved to act as a mass counterbalance. For the serial manipulator, the physical configuration changed to appear as in Figure 2.2.

In Figure 2.2, the two new lengths introduced, l_1' and l_2' are the lengths the links had to be extended to put the motors in the location necessary for tuning. These lengths change the tuning conditions to be;

$$m_1 l_{1c} + m_2 l_1 + m_o l_1 - m_1' l_{1c}' - m_m l_1' = 0 \tag{2.18}$$

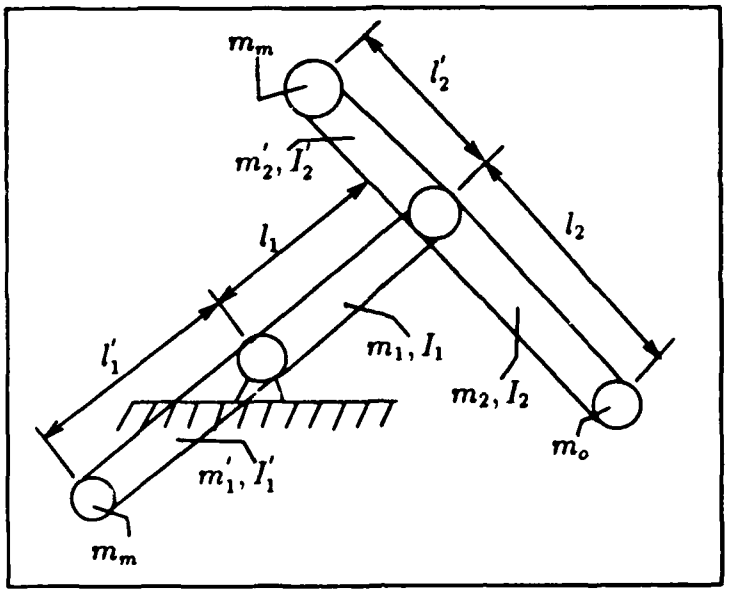

Figure 2.2. Physical Configuration of Tuned Serial Manipulator

and

$$m_1 l_{1c} + m_2 l_1 + m_o l_1 - m_m l_2' - m_2' l_{2c}' = 0 \tag{2.19}$$

The addtional mass m_1' and m_2' is added assuming constant cross sectional area and constant density of the material. Therefore, these masses can be expressed mathematically in terms of the additional length added. Therefore Equations 2.18 and 2.19 are a system of two equations and two unknowns, l_1' and l_2'. Solving the equations, the lengths are:

- $l_1' = 0.4483 meters$
- $l_2' = 0.1455 meters$

This modified configuration was used in the efficiency calculations for the plots in Chapter III.

2.6.5 Parallel Manipulator. The tuning conditions for the parallel manipulator are obtained from Equation 2.9. The parallel manipulator can be made

completely decoupled, invariant, and gravity compensated. To achieve this tuning, three conditions must be met to balance the centers of gravity of the links. These conditions are as follows:

For configuration invariance and dynamic decoupling,

$$m_3 l_2 l_{3c} - m_4 l_1 l_{4c} - m_o l_1 l_4 = 0 \tag{2.20}$$

For gravity compensation,

$$m_1 l_{1c} + m_3 l_{3c} + m_4 l_1 + m_o l_1 = 0 \tag{2.21}$$

$$m_2 l_{2c} + m_3 l_2 - m_4 l_{4c} - m_o l_4 = 0 \tag{2.22}$$

For the parallel manipulator, configuration invariance can be achieved independent of gravity compensation and vice versa. The physical configuration of the parallel arm was changed, consistent with the way the serial was changed, by using the location of the motors to achieve the conditions necessary for configuration invariance and gravity compensation. The physical appearance of this configuration is shown in Figure 2.3.

The two new lengths defined in the physical alteration process for the parallel arm were, l_1' and l_2'. Where l_1' is additional link length added for tuning, and l_2' designates the new location of the motor driving link 2. These new parameters change the tuning constraint equations for gravity compensation of the parallel manipulator. These equations are:

$$m_1 l_{1c} + m_3 l_{3c} + m_4 l_1 + m_o l_1 - m_m l_1' - m_1' l_{1c}' = 0 \tag{2.23}$$

$$m_2 l_{2c} + m_3 l_2 - m_4 l_{4c} - m_o l_4 + m_m l_2' = 0 \tag{2.24}$$

Substituting for m_1' in terms of length l_1', the equations were solved for l_1' and l_2'. For the case of imposing gravity compensation only without configuration invariance, l_{4c} was assumed to be $l_4/2$. Applying the gravity compensation constraints yielded:

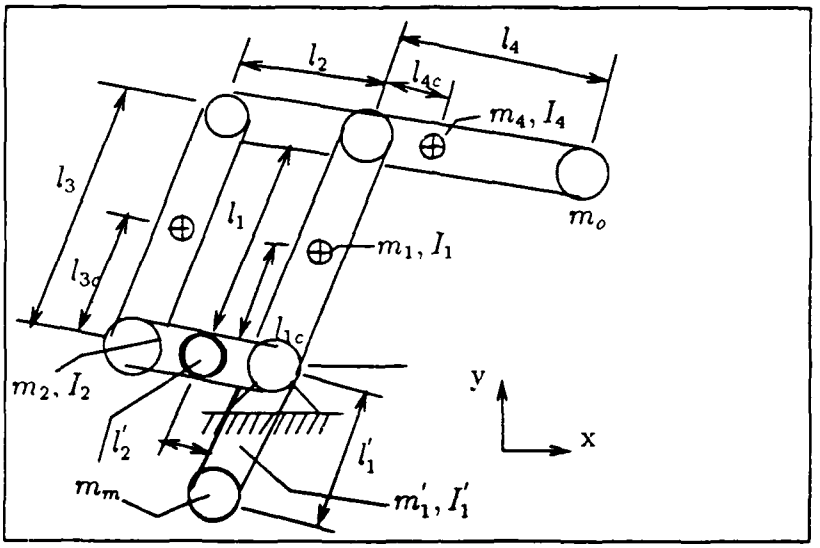

Figure 2.3. Physical Configuration of Tuned Parallel Configuration

- $l_1' = 0.0895\ meters$
- $l_2' = 0.0466\ meters$

When imposing the configuration invariance constraint along with the gravity compensation constraints, the center of gravity of link 4 had to changed. Through redistribution of the mass of link 4, the new center of gravity was driven to $l_{4c} = 0.0629\ meters$ for configuration invariance. The new locations of the motors to achieve gravity compensation became:

- $l_1' = 0.0895\ meters$
- $l_2' = 0.0419\ meters$

This modified configuration was used in the efficiency calculations for the plots in Chapter III.

2.7 *Mechanical Efficiency Comparison*

Although there has been much attention given to how to build and tune manipulators to achieve configuration invariance, dynamic decoupling and gravity compensation, very little attention has been given to evaluating and comparing the performance of these manipulators. Most of the literature [1], [13], addresses how to tune for a constant payload, and claims that payload, indeed, doesn't vary appreciably. However, when considering the most general manipulator, where the application or task of that manipulator is not well defined, the effects of a varying payload on performance are of great concern. Song and Lee, [14], present a method for evaluating the performance of manipulators using mechanical efficiency. Song and Lee, [14], considered only massless systems with point mass payload. The research presented here expands on Song and Lee's work breaking new ground to include inertial and gravitational terms. Coriolis and centrifugal effects have been neglected assuming they are negligible at low speeds.

2.8 *Summary*

This chapter has addressed the design considerations of the parallel and serial configurations. The first consideration is the design goal of trying to define and emulate human arm motion. Then the physical layout of each configuration was examined. Finally, the choices made to apply mechanical efficiency analysis to each configuration in a nominal and various partially tuned and fully tuned configurations were stated.

III. Efficiency Analysis

As a starting point, Song and Lee's work, [14], was recreated. In the analysis, the equations of motion were specialized for each configuration to a massless arm moving the same payload. A vertical trajectory was assumed, with constant x and finite y displacement in cartesian coordinates, the inverse kinematic equations, Equations 2.3 and 2.4, were solved for the angular displacement and substituted into the work integral. For this specialized case the work can be found in a closed form solution.

Having reproduced Song and Lee's work, see section 3.2 for detailed analysis, the next step was to add in the mass of the manipulators and look at the performance. The payload was still being displaced by the same amount, but the manipulators were moving their own mass in addition to the payload. A new performance parameter had to be introduced. Both manipulators were moving the same payload by the same amount. The comparison of performance was how much total work had to be done by the manipulators to produce that movement. The parameter η' was introduced as a comparison of that work, where η' is defined as,

$$\eta' = \frac{Input\ Work\ of\ Parallel\ Manipulator}{Input\ Work\ of\ Serial\ Manipulator} \tag{3.1}$$

This parameter can be greater than one, unlike what we are used to seeing in efficiency. When η' is greater than one, the serial manipulator has done less total work to move the payload a prescribed distance than the parallel manipulator. This would imply that the serial manipulator is more efficient for that movement. If η' is less than 1, the parallel manipulator is more efficient. This performance parameter will be used for the remainder of the analysis.

Even with the mass of the manipulator added in, the gravitational terms of the work integral expression were still integrable in a closed form. A closed form solution of the work integral was not possible with the inertial terms included.

A time varying trajectory was assumed in order to find joint position, velocity, and acceleration for the constant x, 0.1 meter change in y displacement. Constant acceleration, a, in the y direction was assumed to yield the following trajectory profile:

$$\ddot{y} = a \tag{3.2}$$

$$\dot{y} = at \tag{3.3}$$

$$y = \frac{1}{2}at^2 \tag{3.4}$$

$$\tag{3.5}$$

Through the manipulator inverse kinematics, Equations 2.3 and 2.4, expressions for joint velocities and accelerations were found.

$$\dot{\theta}_1 = \frac{l_2 sin(\theta_1 + \theta_2)\dot{y}}{l_1 l_2 sin\theta_2} \tag{3.6}$$

$$\dot{\theta}_2 = \frac{[-l_1 sin\theta_1 - l_2 sin(\theta_1 + \theta_2)]\,\dot{y}}{l_1 l_2 sin\theta_2} \tag{3.7}$$

$$\ddot{\theta}_2 = \frac{l_1 sin\theta_1 \ddot{\theta}_1 + l_1 cos\theta_1 \dot{\theta}_1^2 + l_2 sin(\theta_1 + \theta_2)\ddot{\theta}_1 + l_2 cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)^2}{-l_2 sin(\theta_1 + \theta_2} \tag{3.8}$$

$$\begin{aligned}\ddot{\theta}_1 = {} & \{\ddot{y} + l_2 sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)^2 + [l_1 cos\theta_1 \dot{\theta}_1^2 + l_2 cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)^2] \\ & \times cot(\theta_1 + \theta_2) + l_1 sin\theta_1 \dot{\theta}_1^2\} \\ & \div \{l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) - cot(\theta_1 + \theta_2)[l_1 sin\theta_1 + l_2 sin(\theta_1 + \theta_2)]\}\end{aligned} \tag{3.9}$$

Because of the inability to integrate these equations into a closed form solution of mechanical efficiency, numerical integration was done when the inertial terms were included. The numerical integration was based upon a standard trapezoidal rule, [5, page 777], whose accuracy was confirmed by comparing the numerically integrated gravitational term results to those of the closed form solution. The integration confirmed accuracy to three significant figures. The detailed MatrixX , [16], program developed for this numerical analysis is included in appendix B. The expressions for $\dot{\theta}_1$, $\dot{\theta}_2$, $\ddot{\theta}_1$, and $\ddot{\theta}_2$ and the equations of motion of the serial

and parallel manipulators were evaluated at every step (twenty steps were used to achieve three significant figure accuracy) in the integration. The efficiency ratio was determined for over 750 points in the workspace. With these large expressions being calculated, and considering the intense number of iterations required, the numerical process of evaluating efficiency when including inertial terms became very time consuming.

Once the method of numerically integrating the work integral expressions was checked out for the analysis, the efficiencies, η', were computed for the following cases of manipulator dynamics represented by:

- Only including gravitational terms,
  - Massless configurations,
  - All mass included configurations,
- Inertia terms only,
  - Without configuration invariance,
  - With configuration invariance,
- Inertia and gravity terms included,
  - No constraints imposed,
  - Configuration invariance constraints imposed,
  - Gravity balance and configuration invariance constraints imposed,

As stated earlier, the manipulators were tuned for zero payload and the efficiencies were calculated for a payload of 1.6 kg, in order to show the effects of how the manipulators' efficiencies change due to variation in payload.

The problem of how to implement constraints fairly for both manipulators had to be addressed. The decision was made to use the weight of the motors to

balance the manipulators, which is the common practice for robot design, [13]. This decision avoided the problem of having to add mass to each configuration or changing the baseline dimensions except those required to support the relocated motors.

3.1 η' Presentation Method

η' data was compared by graphical means. The plots generated show a three dimensional representation of the efficiency as it varies over the x-y workspace (first quadrant). All efficiencies were calculated for a small change in the y direction of 0.1 meter of the pointmass payload (1.6 kg) located at the end the manipulator arm. Initial evaluations determined that quadrants other than the first quadrant were symmetrically similiar, therefore it was unnecessary to compute efficiency for the remaining quadrants.

The planar workspace of the manipulators is an annular area with inside radius equal to $(l_2 - l_1)$, (0.3073 meters), and outside radius equal to $(l_1 + l_2)$, (1.0337 meters). Because the base cartesian coordinates were being used, the numerical analysis program calculated rectangular sections of the workspace. Shaded areas of the plots where the efficiency was not calculated are set to be zero. For example, in Figure 3.1, a region made up of two rectangular areas is shown to have a zero value.

Portions of this region are reachable by the manipulator, but because a general trend of the efficiency was being sought it was not necessary to compute the efficiency for these areas. The dotted line enclosing the box of the plot is included to give a reference point for the largest magnitude of that graph.

The purpose of these plots is not to be able to interpolate numbers from them, but rather to see the trend of how the efficiency varies over the workspace, and to compare the efficiency of the serial and parallel configurations. Therefore, the actual magnitude of any individual point on the workspace is of lessor im-

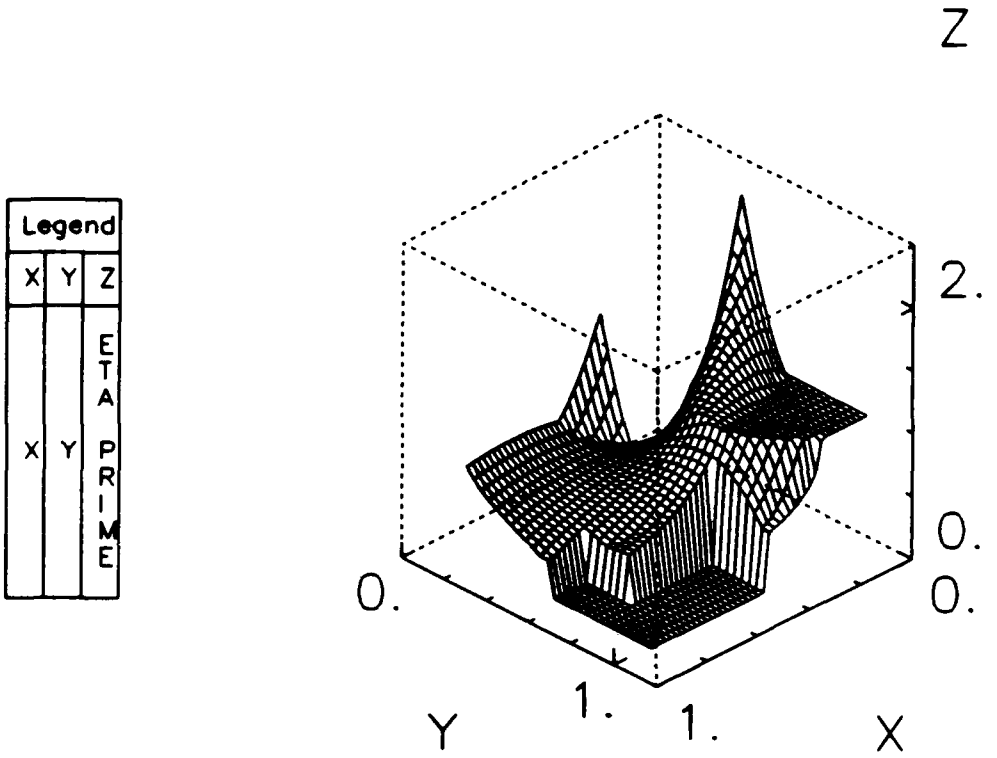

Figure 3.1. Example of an Efficiency Ratio Plot

portantance. The line of demarcation for the plots of η' is $\eta' = 1.0$. When η' is greater than one, the serial arm is more efficient. When η' is less than one, the parallel arm is more efficient. In Figure 3.2, for most of the workspace the value of η' is less than one, indicating a more efficient parallel manipulator. But in one semi-circular region, η' is equal to one, indicating equal efficiency between the two configurations. In some cases, Figure 3.5 for example, the demarcation line is difficult to see because of large magnitude data points causing spikes in the surface generated. When this occurs, an additional plot is included to show how the efficiency behaves over a majority of the workspace showing where the value of one falls. Additional plot generation is done by narrowing the amount of workspace shown, usually part of the y axis, and eliminating the data causing the spiked behavior.

3.2 *Massless Systems*

The first step in this efficiency analysis was to reconfirm the work of Song and Lee, [14]. This was done by integrating the equations of motion to obtain the work. The general expression for work is,

$$W = \int_1^2 \tau \cdot d\theta \tag{3.10}$$

In order to evaluate using the Song and Lee assumptions, the equations of motion had to specialized for a massless system. For the serial manipulator these equations are,

$$\tau_1 = m_o g l_1 cos\theta_1 + m_o l_2 g cos(\theta_1 - \theta_2) \tag{3.11}$$

$$\tau_2 = m_o g l_2 cos(\theta_1 + \theta_2) \tag{3.12}$$

The total work done by the robot is the sum of the absolute values of work done by each actuator. The work done by the joint one actuator of the serial manipulator is,

$$W_{1_s} = \int_{\theta_{1_i}}^{\theta_{1_f}} [m_o g l_1 cos\theta_1 + m_o g l_2 cos(\theta_1 + \theta_2)] d\theta_1 \tag{3.13}$$

The trajectory of the payload in the Song and Lee analysis (and for all efficiencies calculated in this analysis) held x in cartesian coordinates constant. From the forward kinematic equation:

$$x = l_1 cos\theta_1 + l_2 cos(\theta_1 + \theta_2) \tag{3.14}$$

one can solve for $cos(\theta_1 + \theta_2)$ and the integrand is then in terms of the variable θ_1. The integral, evaluated from initial to final values, becomes:

$$W_{1_s} = \int_{\theta_{1_i}}^{\theta_{1_f}} \left[m_o g l_1 cos\theta_1 + m_o g l_2 \left(\frac{x}{l_2} - \frac{l_1}{l_2} cos\theta_1 \right) \right] d\theta_1 \tag{3.15}$$

integrating this expression results in,

$$W_{1_s} = m_o g x [\theta_{1_f} - \theta_{1_i}] \tag{3.16}$$

which is in agreement with Song and Lee, [14]. The expression work of the joint two actuator of the serial manipulator is,

$$W_{2s} = \int_{\theta_{2i}}^{\theta_{2f}} [m_o g l_2 cos(\theta_1 + \theta_2)] d\theta_2 \tag{3.17}$$

To integrate this expression the constraint equation, Equation 3.14, was differentiated and solved for $d\theta_2$.

$$d\theta_2 = \frac{l_1 sin\theta_1}{-l_2 sin(\theta_1 + \theta_2)} d\theta_1 - d\theta_1 \tag{3.18}$$

and from the trigonometric identity $sin^2\theta + cos^2\theta = 1$,

$$sin(\theta_1 + \theta_2) = \sqrt{1 - \left(\frac{x}{l_2} - \frac{l_1}{l_2} cos\theta_1\right)^2} \tag{3.19}$$

Now substituting for $d\theta_2$ and $sin(\theta_1 + \theta_2)$, the integral becomes,

$$W_{2s} = m_o g l_2 \int_{\theta 1i}^{\theta_{1f}} \left[- \frac{\left(\frac{x}{l_2} - \frac{l_1}{l_2} cos\theta_1\right) \frac{l_1}{l_2} sin\theta_1}{\sqrt{1 - \left(\frac{x}{l_2} - \frac{l_1}{l_2} cos\theta_1\right)^2}} - \left(\frac{x}{l_2} - \frac{l_1}{l_2} cos\theta_1\right) \right] d\theta_1 \tag{3.20}$$

Making use of a common substitution, this integral evaluates to,

$$W_{2s} = m_o g l_2 \left\{ [sin(\theta_1 + \theta_2)] \Big|_i^f - \frac{x}{l_2} (\theta_{1f} - \theta_{1i}) + \frac{l_1}{l_2} (sin\theta_{1f} - sin\theta_{1i}) \right\} \tag{3.21}$$

This is also in agreement with Song and Lee, [14].

The equations of motion for the parallel manipulator can be integrated directly and result in the following work expressions,

$$W_{1p} = m_o g l_1 (sin\theta_{1f} - sin\theta_{1i}) \tag{3.22}$$

$$W_{2p} = m_o g l_2 [sin(\theta_{1f} + \theta_{2f}) - sin(\theta_{1i} + \theta_{2i})] \tag{3.23}$$

which again agrees with Song and Lee, [14].

However, Song and Lee make assumptions on signs of the the work expressions and draw conclusions from those assumptions. Song and Lee claim that the parallel manipulator is 100% efficient based on evaluation of one point in the

workspace where the signs of the work expressions agree with their assumptions. However, from this analysis it is seen that the parallel manipulator is not 100% efficient for all the work space. The work done by the two actuators of the parallel manipulator are not always opposite in sign which would lead to %100 efficiency in Song and Lee's analysis, [14]. It can be shown that in some parts of the workspace, particularly the outer edge, that the efficiency of the massless parallel manipulator falls below the %100 percent efficient mark.

The first of the three dimensional plots, Figures 3.2, 3.3, and 3.4 show the mechancial efficiency of the massless serial and parallel configurations for the first quadrant of the robot workspace. Figures 3.2 and 3.3 are plots of the actual efficiency, i.e.

$$\eta = \frac{output\ work}{input\ work} \tag{3.24}$$

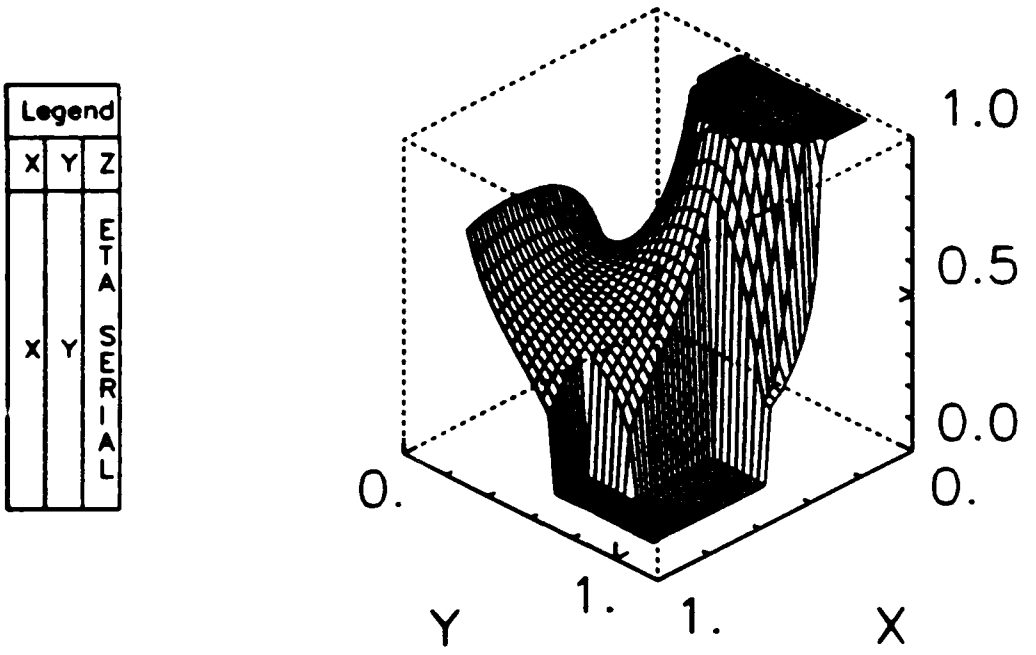

Figure 3.2. Efficiency of Massless Serial Manipulator

and Figure 3.4 is a plot of the comparison of the efficiency ratio,

$$\eta' = \frac{\eta_{serial}}{\eta_{parallel}} \tag{3.25}$$

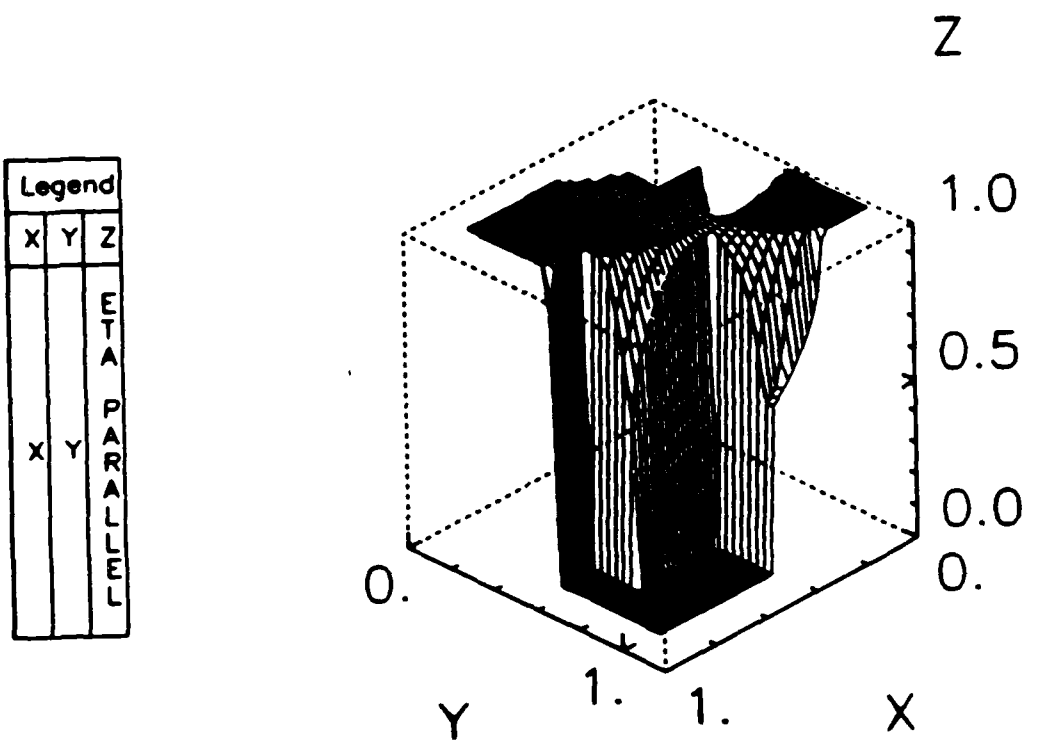

Figure 3.3. Efficiency of Massless Parallel Manipulator

As seen in Figure 3.3, the parallel manipulator is not 100% efficient for all area of the reachable workspace. When the manipulator is near the outer edge of the workspace, near point y=1.0, x=1.0, the efficiency ratio falls below the 100% mark. This less than perfect efficiency is also seen when the manipulator is bent back toward its base, i.e., y=0.3, x=0.0. Figure 3.4 shows the η' plot for the massless case.

This plot shows that the parallel manipulator is more efficient than the serial for the majority of the workspace. The level spot on the graph, located near x=0.0, y=0.5, has a magnitude of 1.0. Except on this plane, the magnitude of η' is mostly less than 1.0. Note that the change in the y direction, used in all the work calculations is 0.1 meters.

This efficiency analysis, Figure 3.4, actually tells us more than just confirming the work of Song and Lee. If the massless manipulators were gravity compensated for some nominal payload, the contribution of the mass of the manipulator and the nominal payload would be eliminated by that compensation. Therefore, these

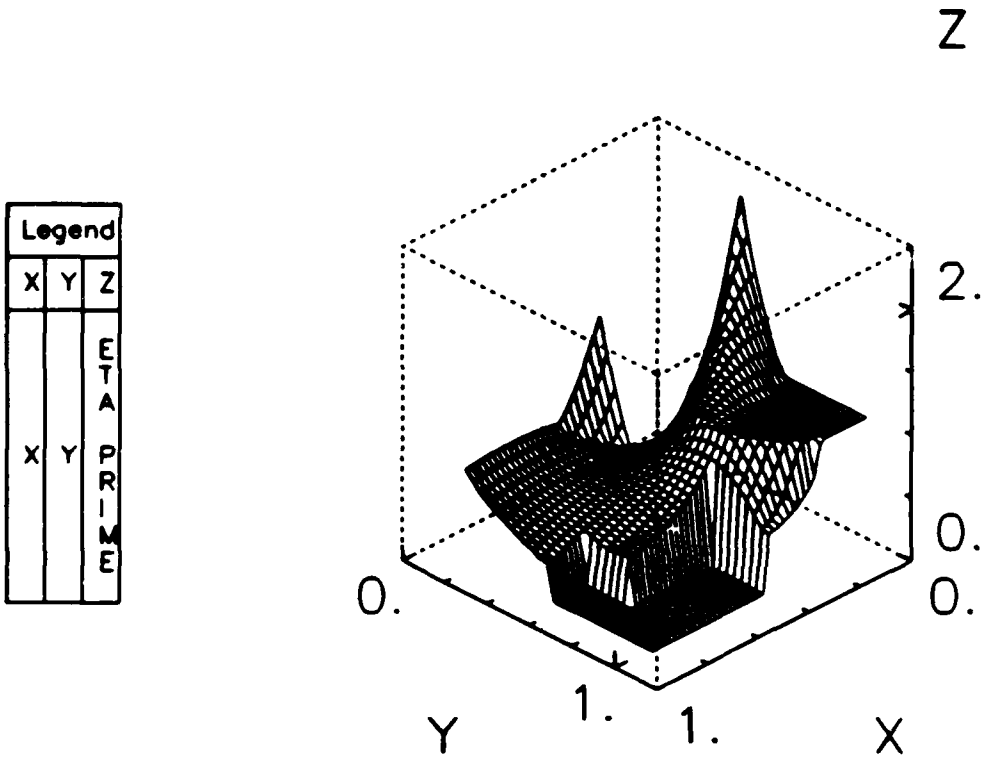

Figure 3.4. η' of Massless Configurations

plots also show how the gravity compensated configurations are effected by some variation in the payload away from the nominal. Note that the efficiency is a function of the location in the workspace.

In general, looking at the η' in Figure 3.4, the total work done by the parallel manipulator, for a delta payload away from some nominal payload used for gravity compensation, is less than the serial manipulator for most of the workspace. Therefore, Song and Lee's conclusion that the parallel configuration is more efficient than the serial configuration is for a general application correct, except when the manipulator is operating in linited parts of workspace.

With Song and Lee's work reproduced and expanded for the entire workspace, the next step was to add in the effects of the mass of the manipulators. For this analysis, the only performance parameter with real meaning is η'.

3.3 *Gravitational Effects*

The next step was to add in the effects of the mass of the configurations. This addition includes the link's mass and inertia. Initially, only gravitational terms of the equations were included. The work expressions were still obtainable in closed form. The integration follows identically. The only changes are the added mass and length terms of the arm, which are a constant to the integration. The work expressions are as follows:

$$W_{1s} = \left(m_1 l_{1c} + m_m l_1 + m_2 l_1 - \frac{m_2 l_{2c} l_1}{l_2}\right)(sin\theta_{1f} - \sin\theta_{1i})\, g + (m_2 l_{2c} + m_o l_2)\frac{gx}{l_2}(\theta_{1f} - \theta_{1i}) \quad (3.26)$$

$$W_{2s} = (m_2 l_{2c} + m_o l_2 g[sin(\theta_1 + \theta_2|_1^2 - \frac{x}{l_2}(\theta_{1f} - \theta_{1i}) + \frac{l_1}{l_2}(sin\theta_{1f} - sin\theta_{1i})] \quad (3.27)$$

$$W_{1p} = (m_1 l_{1c} + m_3 l_{3c} + m_4 l_1 + m_o l_1)\, g\, (sin\theta_{1f} - sin\theta_{1i}) \quad (3.28)$$

$$W_{2p} = (m_2 l_{2c} - m_3 l_2 + m_4 l_{4c} + m_o l_4)\, g[sin\,(\theta_{1f} + \theta_{2f}) - sin(\theta_{1i} + \theta_{2i})] \quad (3.29)$$

Figure 3.5 plots the η' efficiency for the first quadrant of the workspace when only gravitational effects of manipulator mass were included.

As stated in section 3.1, it is difficult to indicate where the demarcation line of when $\eta' = 1.0$ occurs in Figure 3.5. For clarification, Figure 3.6 is included.

Figure 3.6 shows the majority of quadrant one of the workspace with the data causing the spiking effect in Figure 3.5 eliminated. From examining Figure 3.6 one can conclude that when arm gravitational dynamics are included in the η' analysis the parallel configuration is more efficient than the serial configuration for a larger portion of the workspace than when massless configurations were examined. The serial manipulator is still more efficient when the arm is near the point x=0.3,

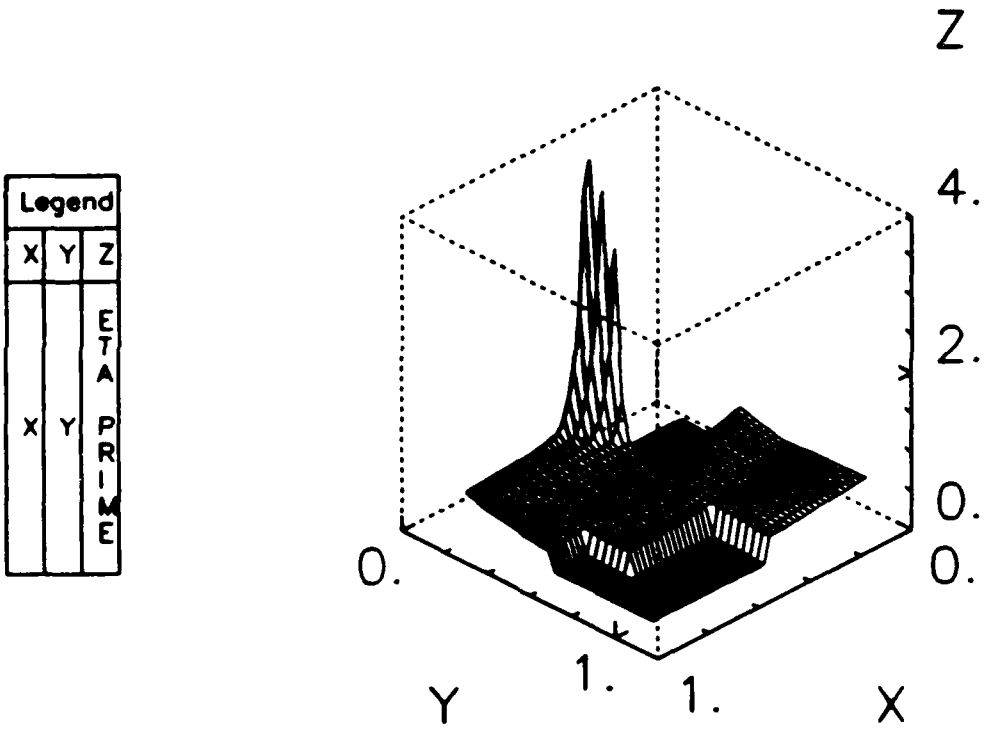

Figure 3.5. Efficiency with Arm Mass, (Gravity Terms Only)

y=0.0. Comparing Figures 3.4 and 3.6, the conclusion is made that significant improvement in efficiency is obtained by applying gravity compensation to either configuration, serial or parallel. However, even with this improvement in efficiency, the parallel configuration is still more efficient, except in a very limited region of the workspace. This completes the computation of gravitational terms. Now, the inertial terms of the equations of motion need to be included.

3.4 Inertial Effects

To determine the work for the the case when inertial terms are included, a closed form solution was not possible. Therefore, numerical integration had to be used. The torques could be calculated from the equations of motion at any given position and acceleration. The acceleration was assumed to be an instantaneous step of magnitude equal to that of gravity in order to compare the magnitude of inertia effects to that of gravity. The numerical integration method used was a trapezoid rule method, which assumes that torque changes linearly between sub-

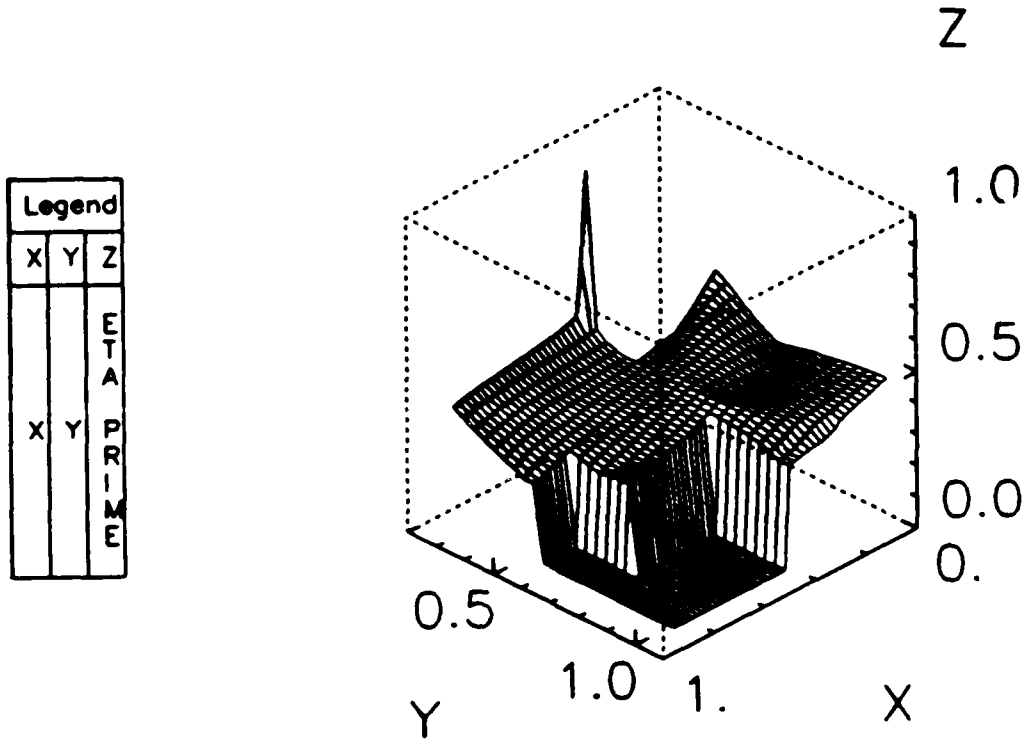

Figure 3.6. Efficiency with Arm Mass, (Gravity Terms Only, Partial Workspace)

sequent positions, [5]. This method was tested by numerically integrating the gravity terms and comparing the result to the closed form solution obtained from that analysis. This method was found to be acccurate to within at least three significant figures.

The efficiency was determined including only inertia terms, without any "tuning", and assuming that the centers of gravity for the links are located half way down the link and are shown in Figure 3.7.

For all of the workspace in Figure 3.7, the parallel configuration is more efficient than the serial when mass effects for only inertia terms are considered. Figure 3.8 represents η' cosidering only inertia terms when the configuration invariance constraints were imposed.

Comparing Figure 3.8 to Figure 3.7, shows that the overall magnitude of η' throughout the workspace has decreased. This indicates the efficiency of the parallel configuration has increased more than that of the serial when configuration

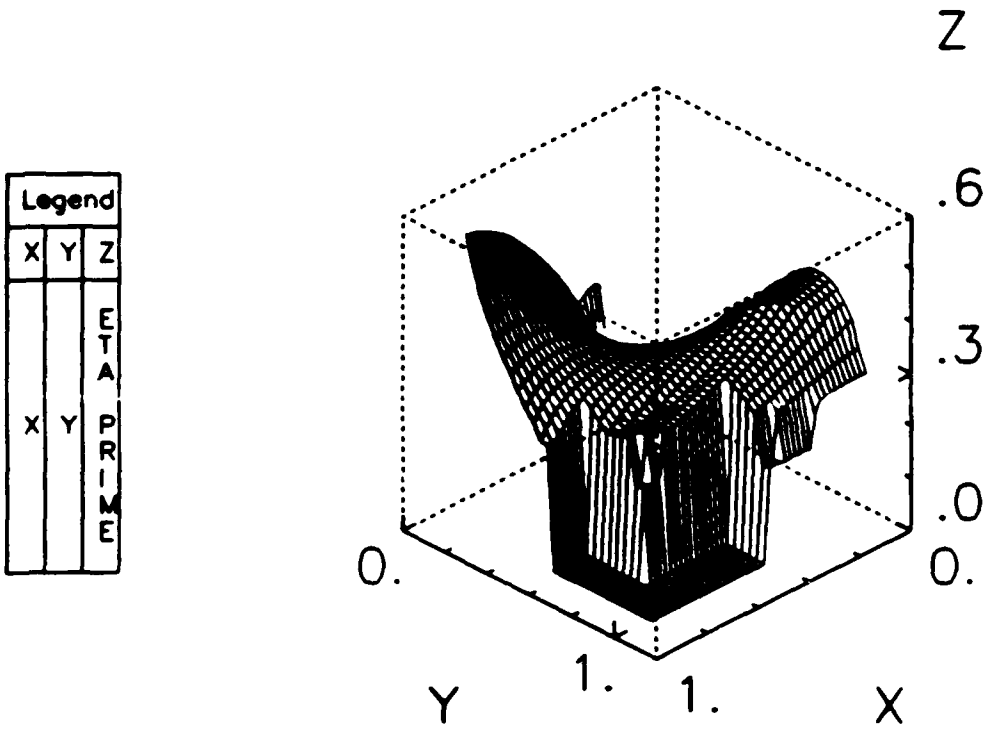

Figure 3.7. Efficiency of Inertia Terms Only, (No Tuning)

invariance is imposed on the inertial terms of the equations of motion. The next step in the analysis is to combine the effects of gravity and inertia, look for the dominate effects, and implement configuration invariance and gravity compensation.

3.5 Combined Effects

For the purpose of this investigation the combined effects refers to the case when both the inertial and gravitational terms of the equations of motion are considered together in the computation of efficiency. The baseline for comparison was the nominal case where no tuning was implemented and all centers of gravity of the links were located at link midpoint. The efficiency plot for this baseline case is in Figure 3.9 and is called the nominal case of the combined efficiency. Again, the line of demarcation between the efficiencies of the two configurations is unclear. Figure 3.10 showing the majority of the workspace and eliminating the spiked behavior is included in order to better visualize the demarcation line.

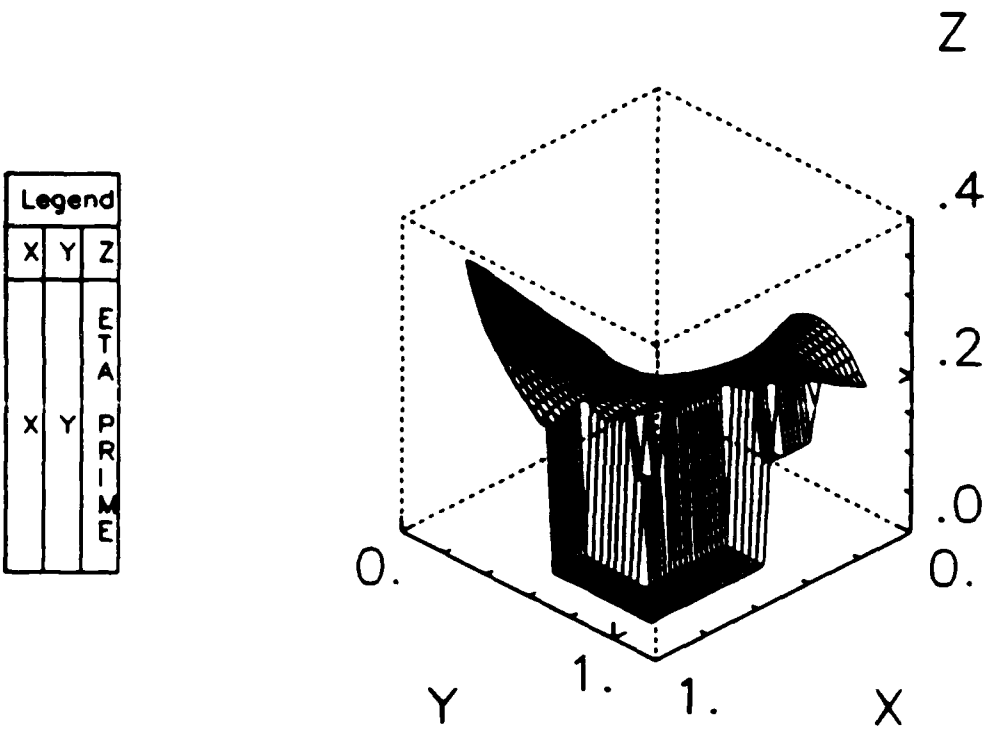

Figure 3.8. Efficiency of Inertia Terms Only, (With Tuning)

Figure 3.10 shows that the parallel configuration is the most efficient for the majority of the workspace. The serial manipulator is more efficient near the point x=0.3, y=0.6, which was previously seen in the case when only gravitational terms were considered.

Using the weights of the motors as counter balances, the link lengths were adjusted to achieve gravity compensation. For the serial manipulator, Figure 1.1, this meant extending link 2 behind the pivotal point (joint 2) by 0.1455 meters, satisfying Equation 2.15. Link 1 was extended by 0.4483 meters to satisfy the condition of equation 2.17. The additional weight added by the extension was added at constant mass per unit length equal to 1.9885 kg/meter. This mass per unit length constant was obtained by using the density of a common aluminum used in manufacturing ($\rho = 2.8 \times 10^3 kg/m^3$) multiplied by the cross sectional area of the links held constant throughout this analysis, see Figure 2.1. The motor mass was considered constant for all motors ($m_m = 6\ kg$).

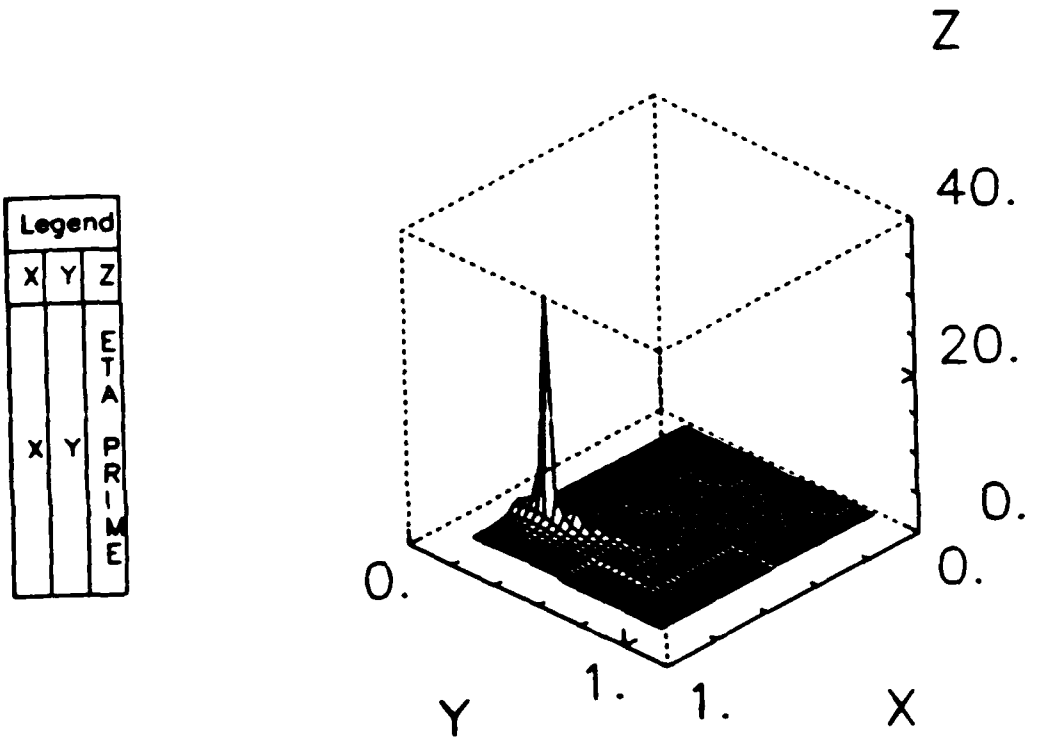

Figure 3.9. Combined Efficiency, (Nominal Case)

For the parallel manipulator, Figure 1.2, link 1 was extended 0.0895 meters to satisfy the gravity balancing constraint, Equation 2.21. To satisfy the constraint for link 2, Equation 2.22, link 2 did not have to be extended, but rather the motor was located 0.0466 meters along the existing link away from the base. With these constraints enforced, the efficiency is plotted in Figure 3.11.

The manipulators are gravity balanced for zero payload, and the efficiency calculated in Figure 3.11, is for a payload of 1.6 kg. Therefore, the plots show effect of a variable payload on the parameter of comparison η'. From Figure 3.11 it is seen that with the improvement afforded by gravity compensation, although the efficiency of both configurations has improved, (see section 3.2), the parallel configuration is the most efficient over all the workspace.

In addition to gravity compensation, the configuration invariance constraint was imposed on the manipulators and the efficiency calculated. This constraint of tuning for configuration invariance is only an additional constraint to the parallel

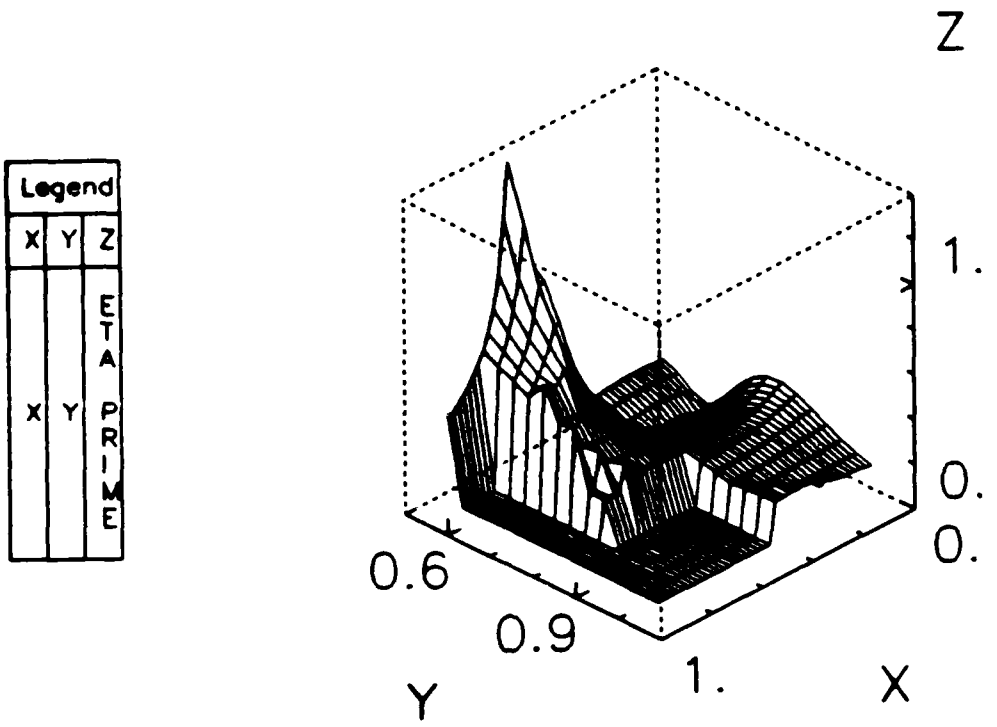

Figure 3.10. Combined Efficiency, (Nominal Case, Partial Workspace)

manipulator, Equation 2.20, because satisfying the gravity balancing constraint for the serial manipulator also gives configuration invariance. With this addition, efficiency was computed and is plotted in Figure 3.12.

Comparing Figures 3.11 and 3.12 it is seen that the magnitude of η' has decreased when configuration invariance was applied to the parallel configuration. The efficiency of the serial arm was unchanged. Looking again at the definition of efficiency used:

$$\eta' = \frac{Input\ Work\ of\ the\ Parallel\ Arm}{Input\ Work\ of\ the\ Parallel\ Arm} \tag{3.30}$$

it is seen the only the work done by the parallel arm has changed. Therefore, if η' decreases in magnitude, the work done by the parallel arm has decreased making it more efficient.

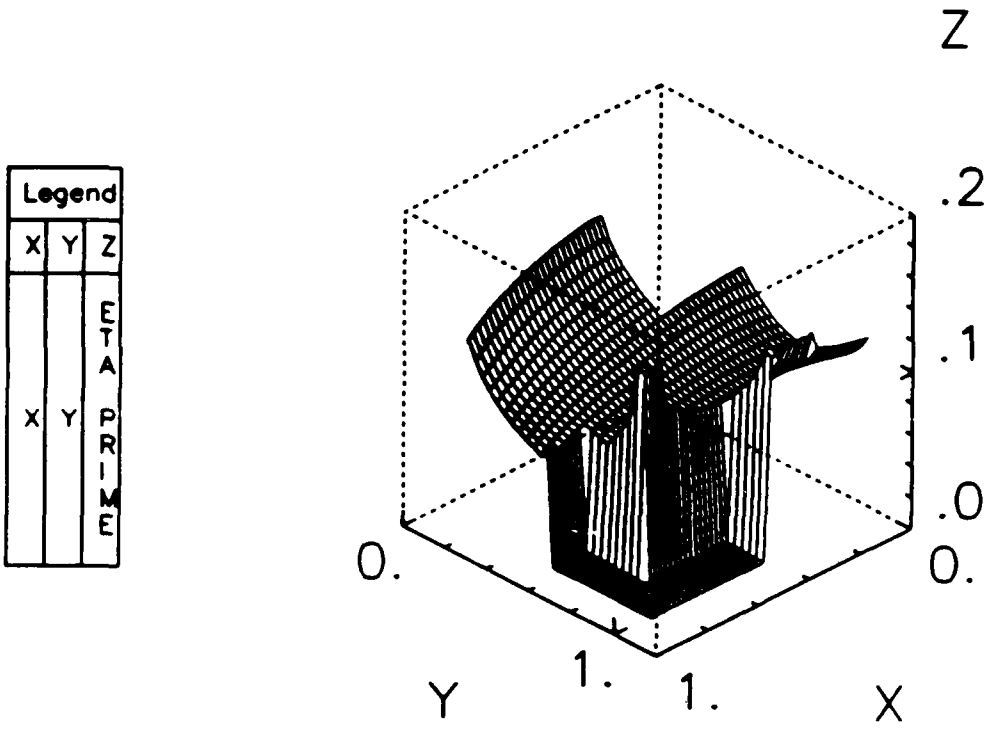

Figure 3.11. Combined Efficiency, Gravity Balanced Case

3.6 Summary

This chapter looked at how the mechanical efficiency was calculated and applied to the parallel and serial robotic configurations. The efficiency was detemined considering the following cases:

- Only gravity terms of the equations of motion.
  - No arm mass included.
  - Arm mass included.
- Inertia terms only.
  - No configuration invariance tuning applied.
  - Configuration invariance tuning applied.
- All terms of the equations of motion included.
  - Nominal case - no tuning.

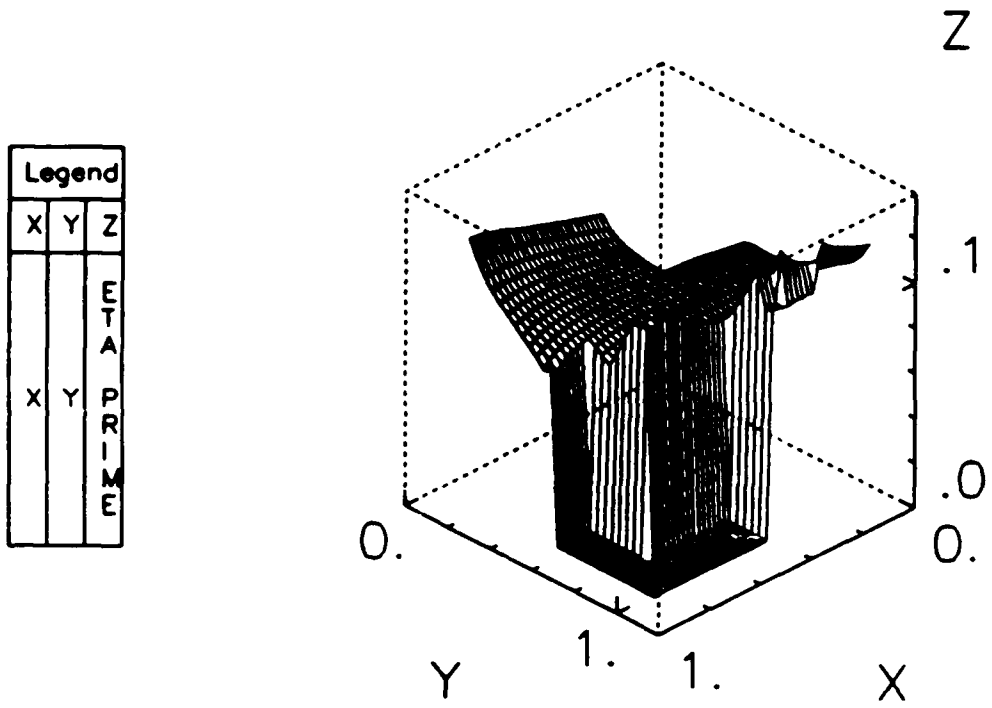

Figure 3.12. Combined Efficiency, Gravity Balanced and Configuration Invariant

- Gravity compensated case.
- Gravity compensated and configuration invariant case.

From the analysis of the data generated, one can draw three basic conclusions.

1. In general the parallel manipulator performs more efficiently than the serial manipulator over a majority of the workspace.
2. Gravity balancing significantly enhances the performance of a manipulator.
3. Trying to achieve minimal control complexity through configuration invariance and dynamic decoupling (achievable only by the parallel manipulator) results in improved efficiency of the manipulator. Miinimal control complexity is not achieved due to the return of non-linear effects arising from carrying a payload other than nominal.

In examination of the plots, Figures 3.2 through 3.12, for the majority of the workspace the efficiency, η', is less than one in value. Looking at the definition

of η', Equation 3.1, this data can be interpreted to mean that the magnitude of the total work done by the parallel manipulator is less than that of the serial manipulator at those points in the work space. Because the points where the value of η' dominate the workspace for all cases considered, it can be stated, by this definition of efficiency, that the parallel manipulator is more efficient. This statement includes the case of both manipulators "tuned" for a specific payload and the actual load was not at that specific value, Figure 3.12.

When comparing gravity compensated data, Figures 3.4 and 3.11, to the uncompensated data, Figures 3.5 and 3.9, that the total work done by both of the manipulators decreases significantly. From this data it can be concluded that, if achievable, gravity compensation is a worthwhile prospect.

Finally, when examining Figures 3.12 and 3.11 one can see that the work done has decreased for the parallel manipulator when making the change to configuration invariance, all other parameters being the same. The serial manipulator parameters were not changed because it achieves configuration invariance when the gravity compensation is applied. Therefore, the lower magnitude of η' can be attributed to the improvement in parallel manipulator efficiency when the conditions for configuration invariance are applied. The conclusion is that minimum control complexity can be achieved, and results in the most efficient configuration.

IV. Conclusions and Recommendations

4.1 Conclusions

A method was developed for determining and comparing the efficiencies of different robotic manipulator configurations. Two robotic manipulator configurations have been compared to determine which would be best suited for application to the A^3RM project. The project requires a manipulator designed for general applications, including the ability to handle variable mass payloads. Current research in efficiency enhancement has generally been applied to manipulator configurations having constant or zero mass payloads. The mechanical efficiency of a open link serial kinematic chain manipulator was compared to the mechanical efficiency of a closed link parallel kinematic chain manipulator considering the effects of carrying a mass payload other than that for which the manipulator was tuned. The results of that comparison are as follows:

1. The parallel manipulator performs more efficiently than the serial over a majority of the workspace.
2. Gravity compensation significantly enhances the efficiency of each manipulator.
3. Through configuration invariance minmal control is achieved and the efficiency of the parallel configuration manipulator is enhanced.

4.2 Recommendations

Although this research is significant, this is only the first step. The technique for evaluation technique is now in place, and further studies in this area could be fruitful.

- A form of the parallel manipulator would be the best choice of the two configurations considered, in terms of efficiency. Consideration of other configurations before making a final decision may be warranted.
- Mass balancing to achieve gravity compensation may not be the best approach. Spring counterbalancing and controlled-force counterbalancing should also be investigated.
- Other trajectories should be investigated, including short trajectories in the x-direction, and long trajectories in both y and x-directions.
- An effort should be made to research other performance measures, such as power based efficiency.
- The next logical extension would be to add in the coriolis and centrifugal terms to this method of analysis, and then extend to three degrees of freedom.
- A study could be done trading off the size of motor used in the efficiency analysis. (i.e. motors of different weight)
- Contours of constant η' in the x-y workspace could be used to better quantify the analysis.
- AFIT would benefit tremendously by having a working copy of the automatic symbolic equations of motion generator in MACSYMA, [15] for the tree structure manipulators. This would benefit instruction, control research, and computer aided design efforts at AFIT.

These are the recommendations for further studies in this area for AFIT. This analysis is very diversified and could be applied in many ways.

Appendix A. *Dynamics*

This appendix looks expressly at the dynamic equations of motion of the manipulator. The approach of using Lagrange multipliers [10] is used to derive these equations for closed link kinematic chain manipulators. The equations are derived symbolically using MACSYMA, an mathematical symbolic manipulation programming tool, developed by M. I. T.,[15], to be able to look at the individual terms of different closed link chain configurations. In examining these individual terms, the manipulator can be designed so that configuration invariance and dynamic decoupling can be achieved [1].

A.1 Symbolic Equations of Motion

Open kinematic chain manipulators have a joint for every degree of freedom in the workspace. Closed kinematic chains have at least one redundant joint effecting the motion of the manipulator. The problem discussed here is one where the closed form solution of the dynamic equations of motion for the closed kinematic chain manipulator is sought. This closed form solution should be expressed in the least number of coordinates as possible, i.e., equal to the number of degrees of freedom. This enables the manipulator to be built without actuation or sensing on the redundant joints.

The method discussed here, for the formulation of the closed form equations, is based on the Lagrangian formulation which has been developed by J. Y. S. Luh and Yuan-Fang Zheng in their paper "Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms", [10]. Using this formulation requires developing the dynamic equations of motion for serial links which is done symbolically using the MACSYMA program implementing the Lagrange-Euler formulation developed by M. C. Leu and N. Hemati. This program is found in Leu and Hemati's paper "Automated Symbolic Derivation of Dynamic Equations of

Motion for Robotic Manipulators", [9], and was implemented at AFIT in February 1988.

Using these two papers as a foundation I then had to add addtional MACSYMA programming to implement the formulation of Luh and Zheng [10] and obtain the symbolic equations of motion for closed kinematic chain manipulators.

A.1.1 *Lagrangian Formulation.* The Lagrangian formulation presented by Luh and Zheng [10] develops the dynamic equations of motion for the closed kinematic chain manipulator using Lagrange multipliers and assuming that each joint senses position, velocity, and acceleration.

Going back to first principles [11], the Lagrangian is defined as:

$$L \equiv T - V \tag{A.1}$$

where T is the total kinetic energy of the system and V is the potential energy of the system. Using Lagrange's variational calculus techniques to develop the equations of motion, the forces are placed into the categories of conservative and non-conservative forces. Lagrange's equations can be written in the form:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_{nc} + \sum_{l=1}^{m} \lambda_l a_{li} \quad (i = 1 \ldots n) \tag{A.2}$$

where Q_{nc} is the non-conservative generalized force, λ_l are the Lagrange multipliers, and the a_{li} are the holonomic constraint equations. The holonomic contraints are obtained from the physical constraint equations through the relationship:

$$a_{li} = \frac{\partial g_l}{\partial q_i} \tag{A.3}$$

where g_l are the physical contraints determined by the system in question. Substituting equation A.3 back into equation A.2 and using vector notation produces the equation consistent with Luh and Zheng [10]:

$$\mathbf{D}\ddot{\vec{q}} + \vec{h} + \vec{g} = \vec{\tau} + \left(\frac{\partial \vec{g}}{\partial \vec{q}}\right)\vec{\lambda} \tag{A.4}$$

where the left hand side of Equation A.4 is from the conservative force side of Lagrange's equations, i.e., the left hand side of Equation A.2, $\vec{\tau}$ is the non-conservative control torque vector, and $\vec{g}$ are the physical constraint equations.

Applying Lagrange's equation, Equation A.4, to the robotic manipulator with n joints (not necessarily independent) results in n equations. But the joint coordinates are related through the m physical constraint equations, i.e. $\vec{g}$. Therefore, the system of equations we have developed is the n Lagrange's equations with n + m unknowns, n unknowns are from the coordinates and m unknowns are from the Lagrange multipliers, and the m physical constraint equations which have the n unknowns of the joint coordinates.

Now the system of equations can be solved.

But these equations apply only to open kinematic chain chain manipulators that are somehow physically constrained. To apply this development to a closed kinematic chain manipulator Luh and Zheng [10]make a virtual cut at a joint in the closed chain that is not actuated. This results in two open kinematicc chains that have a common point at the cut. A position vector from the base to the cut defines that point. This vector is defined by both open chains through a series of homogeneous transformations from the base to the cut. The homogeneous transformation matrix are contructed using the Denavit-Hartenburg representation [3, page 36] and will be of the form:

$$\mathbf{T} = [\vec{n}, \vec{s}, \vec{a}, \vec{p}] \tag{A.5}$$

where $\vec{n}$, $\vec{s}$, and $\vec{a}$, represent the rotation of the coordinate frame of the cut joint from the base coordinate frame, and $\vec{p}$ is the position vector vector from the origin of the base frame to the cut expressed in the base frame coordinate system.

When the virtual cut is done at a revolute joint, the $\vec{p}$ vector and the $\vec{a}$ vector is equal for both open chain transformations. The only difference would be some constant physical offset length between the origins of the coordinate frames at the cut depending on how those coordinate frames are defined.

These two position vectors, $\vec{p}$, give us the physical contraint Equations, $\vec{g}$, required for Lagrange's equations, Equation A.4. The components of the $\vec{p}$ vectors are set equal to each other yielding the m constraint equations.

The next step in the Luh and Zheng application [10] is set the non-conserative torques for unactuated joints equal to zero. The order of the n equations, Equation A.4, is then changed to reflect that the equations with actuated joints appear in the 1 through n-m section of the array, and the equations derived with respect to joints without actuators appear in the n-m+1 through n section of the array. Now the equation array in the n-m+1 through n section can be solved for the Lagrange multipliers:

$$\vec{\lambda} = \left\{ \left[\frac{\partial \vec{g}}{\partial \vec{q}}\right]^{-1} \left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\vec{q}}}\right) - \frac{\partial L}{\partial \vec{q}}\right] \right\}_{(n-m+1)\ thru\ n} \tag{A.6}$$

Having solved for the Lagrange multipliers, these expression are then substituted into the 1 through n equations of motion resulting in an expression for the control torques of the actuated joints, $\vec{\tau}$. This expression can be written as:

$$\vec{\tau} = \left\{ \left[\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\vec{q}}}\right) - \frac{\partial L}{\partial \vec{q}}\right] - \left(\frac{\partial \vec{g}}{\partial \vec{q}}\right) \vec{\lambda} \right\}_{1\ thru\ (n-m)} \tag{A.7}$$

where $\vec{\tau}$ are the generalized or "constrained" equations of motion found in Luh and Zheng [10]. This is a system of n-m equations of motion with n joint coordinate unknowns. The dependent coordinates, n-m of them, can be solved for from the constraint equations, $\vec{g}$. Another method for solving equation A.7 would be to sense position, velocity, and acceleration on all joints.

The ultimate closed form solution would be obtained by eliminating the dependent coordinates. This is done by solving the constraint equations for the dependent coordinates, their rates, and their accelerations and substituted into Equation A.7.

A.1.2 *Implementation.* To apply the Lagrangian formulation presented by Luh and Zheng [10], the Leu and Hemati [9] MACSYMA program was used to symbolically derive the equations of motion for the open kinematic chains. The program is included as follows:

```
/* This program derives the equation of motion of a manipulator
   link using the Lagrangian formulation.
  INPUT ----->
       DOF :  NO. OF DEGREES OF FREEDOM.
       JOINT : THE TYPE OF JOINT (0 FOR REVOLUTE,
                       1 FOR PRISMATIC).
       d,a,alpha :  LINK GEOMETRIC PARAMETERS.
       R :  LINK MASS CENTER POSITION VECTOR.
       M :  LINK MASS.
       MOM :  LINK PSEUDO INERTIA MATRIX.
       LNK :  LINK NUMBER.
       GF :  GRAVITATIONAL FIELD VECTOR.
 OUTPUT ----->
       F[I] :  GENERALIZED FORCE AT JOINT I*/

TMATRIX() := (
  PRINT(" ENTER THE NUMBER OF DEGREES OF FREEDOM"),DOF:READ(),
  FOR I THRU DOF DO (
    PRINT("TYPE 0 IF JOINT IS REVOLUTE AND 1 IF JOINT IS PRISMATIC"),
    PRINT(" "),JOINT:READ(),
    IF JOINT=0 THEN (
       PRINT("INPUT THE PARAMETERS OF THE REVOLUTE JOINT:d,a,alpha"),
       PRINT(" "),D[I]:READ(),AD[I]:READ(),ALF[I]:READ(),PRINT(" "),
       A[I]:MATRIX([COS(Q[I]),-SIN(Q[I])*COS(ALF[I]),
```

```
                    SIN(Q[I])*SIN(ALF[I]),AD[I]*COS(Q[I])],
                   [SIN(Q[I]),COS(Q[I])*COS(ALF[I]),
                    -COS(Q[I])*SIN(ALF[I]),AD[I]*SIN(Q[I])],
                   [0,SIN(ALF[I]),COS(ALF[I]),D[I]],
                   [0,0,0,1]))

   ELSE(PRINT("INPUT THE PARAMETERS OF THE PRISMATIC JOINT:theta,a,alpha"),
        PRINT(" "),TH[I]:READ(),AD[I]:READ(),ALF[I]:READ(),PRINT(" "),
        A[I]:MATRIX([COS(TH[I]),-SIN(TH[I])*COS(ALF[I]),
                     SIN(TH[I])*SIN(ALF[I]),AD[I]*COS(TH[I])],
                    [SIN(TH[I]),COS(TH[I])*COS(ALF[I]),
                     -COS(TH[I])*SIN(ALF[I]),AD[I]*SIN(TH[I])],
                    [0,SIN(ALF[I]),COS(ALF[I]),Q[I]],
                    [0,0,0,1]))),

/* GENERATE THE T MATRICES */
FOR I THRU DOF DO(
    IF I=1 THEN T[I]:A[I] ELSE T[I]:T[I-1].A[I]));

/* TAKE THE FIRST DERIVATIVE OF THE
   T MATRICES W.R.T. THE JOINT VARIABLES.*/
DIFFT1() := (FOR I THRU DOF DO(
                FOR J THRU DOF DO(
            IF I>=J THEN(U[I,J] : DIFF(T[I],Q[J])))));

/* TAKE THE SECOND DERIVATIVE OF THE T MATRICES */
DIFFT2() := (FOR I THRU DOF DO(
                FOR J THRU DOF DO(
            IF I>=J THEN(
               FOR K:J THRU DOF DO(
                  IF I>=K THEN(W[I,J,K] : DIFF(U[I,J],Q[K])))))));

/* INPUT THE MASS PROPERTIES */
INERTIA() := (FOR I THRU DOF DO(
```

```
          PRINT("ENTER THE INERTIA MATRIX FOR LINK NO.  ",I),
                MOM[I]:ENTERMATRIX(4,4),PRINT(" "),
                PRINT("ENTER THE CENTER OF MASS VECTOR FOR LINK NO.  ",I),
          R[I]:ENTERMATRIX(4,1),PRINT(" ")),
                PRINT("ENTER THE GRAVITY FIELD VECTOR"),
                GF:ENTERMATRIX(4,1));

/* DERIVE THE DI TERMS */
TERMDI() := (PRINT(" "),PRINT("ENTER THE LINK NO.  "),LNK:READ(),
             DI : 0,I:LNK,
             FOR PPI:I THRU DOF DO (
             DI : DI + (((-M[PPI]*TRANSPOSE(GF)).U[PPI,I]).R[PPI])),
             DD[I] : DI);

/* DERIVE THE DIJ TERMS */
TERMDIJ() := (I:LNK, L:1,
              FOR J:I THRU DOF DO (
           TRAC : 0,MAXIJ:J,
           FOR P:MAXIJ THRU DOF DO (
              JTQI[P] : MOM[P].TRANSPOSE(U[P,I]),
              FOR L THRU 4 DO (
                 TRAC1 : TRAC + ROW(U[P,J],L).COL(JTQI[P],L),
                 TRAC : TRAC1)),
                    DIJ[I,J] : TRAC));

/* DERIVE THE DIJK TERMS */
TERMDIJK() := (I:LNK, LL:1,
               FOR J THRU DOF DO (
          FOR K:J THRU DOF DO (
            IF J=I AND I >= K THEN
               DK[I,J,K] : 0
                  ELSE(IF (J<I AND J<K AND K>=I) OR (J>=I OR J>=K) THEN(
                        IF I>K THEN
                     (IF I>J THEN MAXIJK:I ELSE MAXIJK:J)
```

```
                    ELSE MAXIJK:K,
                TRACEP : 0,
                FOR PP:MAXIJK THRU DOF DO (
                   JTPI[PP] : MOM[PP].TRANSPOSE(U[PP,I]),
                   FOR LL THRU 4 DO (
                      TRACEP : TRACEP +
                         ROW(W[PP,J,K],LL).COL(JTPI[PP],LL))),
                     DK[I,J,K] : TRACEP)))));

/* COLLECT THE DI, DIJ, DIJK TERMS TO OBTAIN
   THE EQUATION OF MOTION OF LINK I.            */
   FI() := (TRMDIJ : 0.,TRMDIJK : 0.,
            FOR J THRU DOF DO(
               (IF J<LNK THEN
                 TRMDIJ : TRMDIJ + DIJ[J,LNK]*DDQ[J]
                ELSE
              TRMDIJ : TRMDIJ + DIJ[LNK,J]*DDQ[J]),
                FOR K:J THRU DOF DO(
                 IF K=J THEN
                      TRMDIJK : TRMDIJK + DK[LNK,J,K]*DQ[J]*DQ[K]
                 ELSE IF J<LNK AND J<K THEN
                    (IF K>=LNK THEN
                       TRMDIJK : TRMDIJK + 2*DK[LNK,J,K]*DQ[J]*DQ[K]
                     ELSE
                       TRMDIJK : TRMDIJK - 2*DK[K,J,LNK]*DQ[J]*DQ[K])
                  ELSE
                    TRMDIJK : TRMDIJK + 2*DK[LNK,J,K]*DQ[J]*DQ[K])),
            F[LNK] : TRMDIJ + IA[LNK]*DDQ[LNK] + TRMDIJK + DD[LNK]);
```

Having the open kinematic chain dynamic equations of motion, additional MACSYMA programming was necessary to:

1. Isolate the physical constraint equations

2. Find the partial derivatives of the constraint equations, i.e., find the holonomic constraints

3. Assign the unactuated forces to the proper element position in the force vector according to the Luh and Zheng paper [10]

4. Solve for the constrained equations of motion (eliminating the Lagrange multipliers, but still working with an excessive number of coordinates).

Finding the constrained forces can be done in a general sense as demonstrated in Luh and Zheng [10]. I have modified the Leu and Hemati program [9] to solve for the symbolic equations of motion of the closed link kinematic chain manipulator with the stipulation that the virtual cut in the formulation be made at a revolute unactuated joint. This program is as follows:

```
/* Modified version of program developed by Leu and Hemati for
   deriveing equations of motion of a manipulator.  The modification
   implements the method developed by Luh and Zheng for deriving
   equations of motion for closed kinematic chains using the method
   of Lagrange multipliers.  The tools developed for finding the constraint
   equations assume that the virtual cut takes place at a revolute joint.*/


/* This program derives the equation of motion of a manipulator
   link using the Lagrangian formulation.
  INPUT ----->
       DOF :  NO. OF DEGREES OF FREEDOM.
       JOINT : THE TYPE OF JOINT (0 FOR REVOLUTE,
                       1 FOR PRISMATIC).
       d,a,alpha :  LINK GEOMETRIC PARAMETERS.
       R :  LINK MASS CENTER POSITION VECTOR.
       M :  LINK MASS.
       MOM :  LINK PSEUDO INERTIA MATRIX.
       LNK :  LINK NUMBER.
       GF :  GRAVITATIONAL FIELD VECTOR.
 OUTPUT ----->
```

```
      F[I] :  GENERALIZED FORCE AT JOINT I*/

TMATRIX() := (
  PRINT(" ENTER THE NUMBER OF DEGREES OF FREEDOM"),DOF:READ(),
  FOR I THRU DOF DO (
    PRINT("TYPE 0 IF JOINT IS REVOLUTE AND 1 IF JOINT IS PRISMATIC"),
    PRINT(" "),JOINT:READ(),
    PRINT("INPUT THE DESIRED JOINT NUMBER"), /*modification*/
    PRINT(" "), DESIRED[I]: READ(),    /*mod*/
    Q[I]:Q[DESIRED[I]],   /*mod*/
    IF JOINT=0 THEN (
       PRINT("INPUT THE PARAMETERS OF THE REVOLUTE JOINT:d,a,alpha"),
       PRINT(" "),D[I]:READ(),AD[I]:READ(),ALF[I]:READ(),PRINT(" "),
       A[I]:MATRIX([COS(Q[I]),-SIN(Q[I])*COS(ALF[I]),
              SIN(Q[I])*SIN(ALF[I]),AD[I]*COS(Q[I])],
                   [SIN(Q[I]),COS(Q[I])*COS(ALF[I]),
              -COS(Q[I])*SIN(ALF[I]),AD[I]*SIN(Q[I])],
             [0,SIN(ALF[I]),COS(ALF[I]),D[I]],
             [0,0,0,1]))

    ELSE(PRINT("INPUT THE PARAMETERS OF THE PRISMATIC JOINT:theta,a,alpha"),
         PRINT(" "),TH[I]:READ(),AD[I]:READ(),ALF[I]:READ(),PRINT(" "),
         A[I]:MATRIX([COS(TH[I]),-SIN(TH[I])*COS(ALF[I]),
                SIN(TH[I])*SIN(ALF[I]),AD[I]*COS(TH[I])],
                     [SIN(TH[I]),COS(TH[I])*COS(ALF[I]),
                -COS(TH[I])*SIN(ALF[I]),AD[I]*SIN(TH[I])],
               [0,SIN(ALF[I]),COS(ALF[I]),Q[I]],
               [0,0,0,1]))),

/* GENERATE THE T MATRICES */
FOR I THRU DOF DO(
    IF I=1 THEN T[I]:A[I] ELSE T[I]:T[I-1].A[I]));

/* TAKE THE FIRST DERIVATIVE OF THE
```

```
   T MATRICES W.R.T. THE JOINT VARIABLES.*'
DIFFT1() := (FOR I THRU DOF DO(
                FOR J THRU DOF DO(
            IF I>=J THEN(U[I,J] : DIFF(T[I],Q[J])))));

/* TAKE THE SECOND DERIVATIVE OF THE T MATRICES */
DIFFT2() := (FOR I THRU DOF DO(
                FOR J THRU DOF DO(
            IF I>=J THEN(
               FOR K:J THRU DOF DO(
                  IF I>=K THEN(W[I,J,K] : DIFF(U[I,J],Q[K])))))));

/* INPUT THE MASS PROPERTIES */
INERTIA() := (FOR I THRU DOF DO(
          PRINT("ENTER THE INERTIA MATRIX FOR LINK NO.  ",I),
               MOM[I]:ENTERMATRIX(4,4),PRINT(" "),
          PRINT("ENTER THE CENTER OF MASS VECTOR FOR LINK NO.  ",I),
          R[I]:ENTERMATRIX(4,1),PRINT(" ")),
             PRINT("ENTER THE GRAVITY FIELD VECTOR"),
             GF:ENTERMATRIX(4,1));

/* DERIVE THE DI TERMS */
TERMDI() := (PRINT(" "),PRINT("ENTER THE LINK NO.  "),LNK:READ(),
             DI : 0,I:LNK,
             FOR PPI:I THRU DOF DO (
          DI : DI + (((-M[DESIRED[PPI]]*TRANSPOSE(GF)).U[PPI,I]).R[PPI])),
             DD[I] : DI);

/* DERIVE THE DIJ TERMS */
TERMDIJ() := (I:LNK, L:1,
                FOR J:I THRU DOF DO (
            TRAC : 0,MAXIJ:J,
            FOR P:MAXIJ THRU DOF DO (
               JTQI[P] : MOM[P].TRANSPOSE(U[P,I]),
```

```
                FOR L THRU 4 DO (
                   TRAC1 : TRAC + ROW(U[P,J],L).COL(JTQI[P],L),
                   TRAC : TRAC1)),
                      DIJ[I,J] : TRAC));

/* DERIVE THE DIJK TERMS */
TERMDIJK() := (I:LNK, LL:1,
                 FOR J THRU DOF DO (
           FOR K:J THRU DOF DO (
             IF J=I AND I >= K THEN
                DK[I,J,K] : 0
                   ELSE(IF (J<I AND J<K AND K>=I) OR (J>=I OR J>=K) THEN(
                         IF I>K THEN
                      (IF I>J THEN MAXIJK:I ELSE MAXIJK:J)
                         ELSE MAXIJK:K,
                    TRACEP : 0,
                    FOR PP:MAXIJK THRU DOF DO (
                       JTPI[PP] : MOM[PP].TRANSPOSE(U[PP,I]),
                       FOR LL THRU 4 DO (
                          TRACEP : TRACEP +
                             ROW(W[PP,J,K],LL).COL(JTPI[PP],LL))),
                      DK[I,J,K] : TRACEP)))));

/* COLLECT THE DI, DIJ, DIJK TERMS TO OBTAIN
   THE EQUATION OF MOTION OF LINK I.            */
   FI() := (TRMDIJ : 0.,TRMDIJK : 0.,
            FOR J THRU DOF DO(
               (IF J<LNK THEN
           TRMDIJ : TRMDIJ + DIJ[J,LNK]*DDQ[DESIRED[J]]
                ELSE
           TRMDIJ : TRMDIJ + DIJ[LNK,J]*DDQ[DESIRED[J]]),
                FOR K:J THRU DOF DO(
            IF K=J THEN
              TRMDIJK : TRMDIJK + DK[LNK,J,K]*DQ[DESIRED[J]]*
```

```
            DQ[DESIRED[K]]
                ELSE IF J<LNK AND J<K THEN
             (IF K>=LNK THEN
                TRMDIJK : TRMDIJK + 2*DK[LNK,J,K]*
                DQ[DESIRED[J]]*DQ[DESIRED[K]]
                    ELSE
                TRMDIJK : TRMDIJK - 2*DK[K,J,LNK]*
                 DQ[DESIRED[J]]*DQ[DESIRED[K]])
                ELSE
              TRMDIJK : TRMDIJK + 2*DK[LNK,J,K]*
              DQ[DESIRED[J]]*DQ[DESIRED[K]])),
          F[DESIRED[LNK]] : [illegible]DIJ + IA[DESIRED[LNK]]*
                           DDQ[DESIRED[LNK]] + TRMDIJK + DD[LNK]);

/* END OF THE LEU AND HEMATI MODIFIED CODE, BEGINNING OF CODE DEVELOPED
   SPECIFICALLY FOR THE FORMULATION OF THE CLOSED CHAIN EOMS */

/* TOOL TO SAVE THE TMATRIX FROM THE BASE TO THE VIRTUAL CUT
   FOR EACH SERIAL CHAIN */

TCUT():= (PRINT("INPUT JOINT NUMBER AT CUT JOINT"),
          PRINT(" "), I:READ(),
          TCUT[I]: T[DOF]);

/* TOOL FOR CALCULATION OF ALL POSSIBLE CONSTRAINT EQUATIONS.
   USER IS RESPONSIBLE FOR FORMING CONSTRAINT MATRIX WITH THE
   INDEPENDENT CONSTRAINTS  */

CONSTRAINTS():= (PRINT("INPUT JOINT NUMBERS OF CONNECTING JOINTS: i, j"),
          PRINT(" "), i: READ(), j: READ(),
          PRINT("TYPE 0 IF THE JOINT IS REVOLUTE AND 1 IF PRISMATIC"),
          PRINT(" "), JOINT: READ(),
          IF JOINT= 0 THEN (
             PRINT("INPUT THE JOINT OFFSET: d"),
```

```
            PRINT(" "), d: READ(),
            C[1]: TCUT[i][1,4]-TCUT[j][1,4],
            C[2]: TCUT[i][2,4]-TCUT[j][2,4],
            C[3]: TCUT[i][3,4]-TCUT[j][3,4] - d,
            C[4]: TCUT[i][1,3]-TCUT[j][1,3],
            C[5]: TCUT[i][2,3]-TCUT[j][2,3],
            C[6]: TCUT[i][3,3]-TCUT[j][3,3])
               ELSE (PRINT("INPUT THE OFFSET ANGLE: THETA"),
              PRINT(" "), THETA: READ(),
              C[1]: TCUT[i][4,1]-TCUT[j][4,1],
              C[2]: TCUT[i][4,2]-TCUT[j][4,2],
                    C[3]: TCUT[i][3,1]-TCUT[j][3,1],
              C[4]: TCUT[i][3,2]-TCUT[j][3,2],
              C[5]: TCUT[i][3,3]-TCUT[j][3,3],
              C[6]: TRANSPOSE(COLUMN(TCUT[i],1)).
                    COLUMN(TCUT[j],1) - THETA));


/* TOOL TO FIND THE HOLONOMIC CONSTRAINT EQUATIONS, THIS IS ONLY FOR
   THE CASE WHERE THE VIRTUAL CUT IS A REVOLUTE JOINT*/

HOLCONSTRAINTS():= (
dc[1,1]:diff(c[1],q[1]),
dc[1,2]:diff(c[2],q[1]),
dc[2,1]:diff(c[1],q[2]),
dc[2,2]:diff(c[2],q[2]),
dc[3,1]:diff(c[1],q[3]),
dc[3,2]:diff(c[2],q[3]),
dc[4,1]:diff(c[1],q[4]),
dc[4,2]:diff(c[2],q[4]))$

/* TOOL TO REORDER THE EQUATIONS OF MOTION IN LUH AND ZHENG FASHION */

REORDEREOM():= (
```

```
noforce:matrix([f[3]],[f[4]]),
actforce:matrix([f[1]],[f[2]]))$

/* TOOL TO BREAK HOLONOMIC CONSTRAINTS INTO ACTUATED AND UNACTUATED
    SECTIONS */

DIVIDEHOLONOMICS():= (
cn:matrix([dc[1,1],dc[1,2]],
          [dc[2,1],dc[2,2]]),
cm:matrix([dc[3,1],dc[3,2]],
          [dc[4,1],dc[4,2]]))$


/* FUNCTION TO SOLVE FOR THE LAGRANGE MULTIPLIERS */

LAMBDA():= (
lambda:invert(cm).noforce)$


/* FUNCTION TO SOLVE FOR THE GENERALIZED FORCES */

FORCES():= (
fc:actforce-cn.lambda)$

/* END OF PROGRAM */
```

The last step of reducing the equations of motion to be functions of only the independent coordinates is not readily generalized. The constraint equations include triginometric operations on the coordinates. Therefore, solving for those coordinates results in non-unique solutions, i.e., angles multiplied by factors of 2π.

However, if all the joints are revolute the sines and cosines of the coordinates can be solved for. From these expressions the joint rates and accelrations can be found. Then the equations of motion can be reduced to functions of only their

independent coordinates. This development is best shown by means of an example. Using the manipulator shown in Figure 1.2, the following constraint equations are derived:

The MACSYMA generated equations of motion are shown in the following *script*. *This script* is the actual MACSYMA transaction of the derivation of the equations of motion for the parallel bar closed link configuration.

```
(d2)          pbarcoeffscript

(c3) fc[1]:fullratsubst(1,cos(q[1])**2+sin(q[1])**2,fc[1])$

Batching the file /usr/local/lib/macsyma/share/lrats.mac

(c3) /* -*- MACSYMA -*- */

EVAL_WHEN(BATCH,TTYOFF:TRUE)$

(c3) /*ASB;LRATS 3
5:05pm  Tuesday, 14 July 1981
7:53pm  Saturday, 29 May 1982
  Added a DIAGEVAL_VERSION for this file.
1:43pm  Saturday, 12 June 1982
  Changed loadflags to getversions, DEFINE_VARIABLE:'MODE.
*/

EVAL_WHEN(TRANSLATE,
  DEFINE_VARIABLE:'MODE,
  TRANSCOMPILE:TRUE)$

(c3) PUT('LRATS,3,'DIAGEVAL_VERSION)$

(c3) DEFINE_VARIABLE(MESSLRATS2,"Invalid argument to FULLRATSUBST:",ANY)$

/usr/local/lib/macsyma/transl/trmode.o being loaded.

(c3) DEFINE_VARIABLE(FULLRATSUBSTFLAG,FALSE,BOOLEAN)$

(c3) LRATSUBST(LISTOFEQNS,EXP):=BLOCK(
  [PARTSWITCH:TRUE,INFLAG:TRUE,PIECE],
  IF NOT LISTP(LISTOFEQNS)
  THEN IF INPART(LISTOFEQNS,0)="="
       THEN LISTOFEQNS:[LISTOFEQNS]
```

```
      ELSE IF FULLRATSUBSTFLAG=TRUE
           THEN ERROR(MESSLRATS2,[LISTOFEQNS,EXP])
   ELSE ERROR("Invalid argument to LRATSUBST:",[LISTOFEQNS,EXP]),
  FOR IDUM IN LISTOFEQNS DO
      IF INPART(IDUM,0)#"="
      THEN IF FULLRATSUBSTFLAG=TRUE
           THEN ERROR(MESSLRATS2,[LISTOFEQNS,EXP])
   ELSE ERROR("Invalid argument to LRATSUBST:",[LISTOFEQNS,EXP]),
  LRATSUBST1(LISTOFEQNS,EXP))$

(c3) LRATSUBST1(LISTOFEQNS,EXP):=BLOCK(
  [DUM:IF LISTOFEQNS=[]
       THEN EXP
       ELSE IF REST(LISTOFEQNS)=[]
            THEN RATSUBST(INPART(LISTOFEQNS,1,2),INPART(LISTOFEQNS,1,1),EXP)
            ELSE LRATSUBST1(REST(LISTOFEQNS),
    IF FULLRATSUBSTFLAG=TRUE
    THEN FULLRATSUBST1(INPART(LISTOFEQNS,1,2),
       INPART(LISTOFEQNS,1,1),
       EXP)
    ELSE RATSUBST(INPART(LISTOFEQNS,1,2),
  INPART(LISTOFEQNS,1,1),
  EXP))],
  DECLARE(DUM,SPECIAL),
  IF FULLRATSUBSTFLAG=TRUE AND DUM#EXP
  THEN LRATSUBST1(LISTOFEQNS,DUM)
  ELSE IF DUM#EXP
       THEN DUM
       ELSE EXP)$

(c3) FULLRATSUBST1(SUBSTEXP,FOREXP,EXP):=BLOCK(
  [DUM:RATSUBST(SUBSTEXP,FOREXP,EXP)],
  IF DUM=EXP
  THEN EXP
  ELSE FULLRATSUBST1(SUBSTEXP,FOREXP,DUM))$

(c3) FULLRATSUBST([ARGLIST]):=BLOCK(
  [FULLRATSUBSTFLAG:TRUE,LARGLISTDUM:LENGTH(ARGLIST),FARGLIST,
   PARTSWITCH:TRUE,INFLAG:TRUE,PIECE],
  IF LARGLISTDUM=2
  THEN IF LISTP(FARGLIST:FIRST(ARGLIST)) OR INPART(FARGLIST,0)="="
       THEN LRATSUBST(FARGLIST,LAST(ARGLIST))
```

```
      ELSE ERROR(MESSLRATS2,ARGLIST)
  ELSE IF LARGLISTDUM=3
      THEN APPLY('FULLRATSUBST1,ARGLIST)
      ELSE ERROR(MESSLRATS2,ARGLIST))$

(c3) EVAL_WHEN(BATCH,TTYOFF:FALSE)$

Batching done.

(c4) fc[2]:fullratsubst(1,cos(q[1])**2+sin(q[1])**2,fc[2])$

(c5) fc[1]:fullratsubst(1,cos(q[2])**2+sin(q[2])**2,fc[1])$

(c6) fc[2]:fullratsubst(1,cos(q[2])**2+sin(q[2])**2,fc[2])$

(c7) fc[1]:fullratsubst(dq[1]-dq[2],dq[3],fc[1])$

(c8) fc[2]:fullratsubst(dq[1]-dq[2],dq[3],fc[2])$

(c9) fc[1]:fullratsubst(ddq[1]-ddq[2],ddq[3],fc[1])$

(c10) fc[2]:fullratsubst(ddq[1]-ddq[2],ddq[3],fc[2])$

(c11) fc[1]:fullratsubst(-dq[1]+dq[2],dq[4],fc[1])$

(c12) fc[2]:fullratsubst(-dq[1]+dq[2],dq[4],fc[2])$

(c13) fc[1]:fullratsubst(-ddq[1]+ddq[2],ddq[4],fc[1]);

(d13) [(l  m  + g  m  + g  m ) cos(q ) v
        1  4    3  3    1  1       1

 + ((ddq  l  g  m  - l  ddq  g  m ) sin(q )
       2  2  3  3    1    2  4  4       1

          2           2
 + (l  dq  g  m  - dq  l  g  m ) cos(q )) sin(q )
    1   2  4  4     2  2  3  3       1        2

       2                 2
 + ((dq  l  g  m  - l  dq  g  m ) sin(q )
       2  2  3  3    1   2  4  4       1
```

```
                                                                    2
 + (ddq  l  g  m  - l  ddq  g  m ) cos(q )) cos(q ) + ddq  l  m
      2  2  3  3    1    2  4  4       1        2       1  1  4

                          2                                         2
 +(ddq - ddq )ia +ddq  g  m +ddq izz +(ddq -ddq )ia + ddq  g  m
     1     2   4    1  3  3    1   3     1    2   3     1  1  1

 + ddq  izz  + ddq  ia ]
     1    1      1   1
```

```
(c14) fc[2]:fullratsubst(-ddq[1]+ddq[2],ddq[4],fc[2]);
```

```
(d14) [(- g  m  + l  m  + g  m ) cos(q ) v
          4  4    2  3    2  2       2

 + ((ddq  l  g  m  - ddq  l  g  m ) sin(q )
       1  2  3  3      1  1  4  4       1

     2              2
 + (dq  l  g  m  - dq  l  g  m ) cos(q )) sin(q )
     1  2  3  3     1  1  4  4       1        2

      2              2
 + ((dq  l  g  m  - dq  l  g  m ) sin(q )
      1  1  4  4     1  2  3  3       1

                                                        2
 +(ddq  l  g  m - ddq  l  g  m )cos(q ))cos(q )+ ddq  g  m +ddq  izz
     1  2  3  3     1  1  4  4      1       2      2  4  4    2    4

                          2                                 2
 +(ddq - ddq )ia + ddq  l  m +(ddq -ddq )ia  + ddq  g  m  + ddq  izz
     2     1   4     2  2  3     2    1   3      2  2  2       2    2

 + ddq  ia ]
     2   2
```

```
(c15) fc[1]:fullratsimp(fc[1]);
```

```
(d15) [(l  m  + g  m  + g  m ) cos(q ) v
        1  4    3  3    1  1       1
```

```
+ ((ddq  l  g  m  - l  ddq  g  m ) sin(q )
       2  2  3  3    1    2  4  4       1

        2           2
+ (l  dq  g  m  - dq  l  g  m ) cos(q )) sin(q )
    1   2  4  4     2  2  3  3       1        2

      2                  2
+ ((dq  l  g  m  - l  dq  g  m ) sin(q )
      2  2  3  3    1  2  4  4

                                                          2
+ (ddq  l  g  m  - l  ddq  g  m ) cos(q )) cos(q ) + ddq  l  m
      2  2  3  3    1    2  4  4       1        2       1  1  4

                        2                                   2
+(ddq -ddq )ia +ddq  g  m +ddq izz +(ddq -ddq )ia +ddq  g  m
     1    2   4    1  3  3    1   3     1    2   3    1  1  1

+ ddq  izz  + ddq  ia ]
     1    1      1   1

(c16) fc[2]:fullratsimp(fc[2]);

(d16) [(- g  m  + l  m  + g  m ) cos(q ) v
          4  4    2  3    2  2       2

 + ((ddq  l  g  m  - ddq  l  g  m ) sin(q )
       1  2  3  3      1  1  4  4       1
     2              2
 + (dq  l  g  m  - dq  l  g  m ) cos(q )) sin(q )
     1  2  3  3     1  1  4  4       1        2


      2              2
 + ((dq  l  g  m  - dq  l  g  m ) sin(q )
       1  1  4  4     1  2  3  3       1

                                                    2
 +(ddq  l  g  m -ddq  l  g  m )cos(q ))cos(q )+ ddq  g  m +ddq izz
     1  2  3  3    1  1  4  4      1       2      2  4  4    2   4
```

```
                          2                          2
 +(ddq - ddq )ia +ddq  l  m +(ddq -ddq )ia  + ddq  g  m  + ddq  izz
     2     1   4    2  2  3     2    1   3      2  2  2      2    2

 + ddq  ia ]
     2   2

(c17) d11:ratcoeff(fc[1],ddq[1]);

             2             2                  2
(d17)      [l  m  + ia  + g  m  + izz  + ia  + g  m  + izz  + ia ]
             1  4     4    3  3      3     3    1  1      1     1

(c18) d12:ratcoeff(fc[1],ddq[2]);

(d18) [(l  g  m  - l  g  m ) sin(q ) sin(q )
        2  3  3    1  4  4       1       2

              + (l  g  m  - l  g  m ) cos(q ) cos(q ) - ia  - ia ]
                  2  3  3    1  4  4       1       2      4     3

(c19) d21:ratcoeff(fc[2],ddq[1]);

(d19) [(l  g  m  - l  g  m ) sin(q ) sin(q )
        2  3  3    1  4  4       1       2

       + (l  g  m  - l  g  m ) cos(q ) cos(q ) - ia  - ia ]
           2  3  3    1  4  4       1       2      4     3

(c20) d22:ratcoeff(fc[2],ddq[2]);

       2             2             2
(d20  [g  m  + izz  + ia  + l  m  + ia  + g  m  + izz  + ia ]
       4  4     4     4    2  3     3    2  2      2     2

(c21) g1:ratcoeff(fc[1],v);

(d21)                 [(l  m  + g  m  + g  m ) cos(q )]
                         1  4    3  3    1  1       1

(c22) g2:ratcoeff(fc[2],v);
```

```
(d22)        [(- g  m  + l  m  + g  m ) cos(q )]
                  4  4    2  3    2  2       2

(c23) h111:ratcoeff(fc[1],dq[1],2);

(d23)        [0]

(c24) h112:ratcoeff(fc[1],dq[1],1);

(d24)        [0]

(c25) h122:ratcoeff(fc[1],dq[2],2);

(d25) [(l  g  m  - l  g  m ) cos(q ) sin(q )
        1  4  4    2  3  3       1       2

       + (l  g  m  - l  g  m ) sin(q ) cos(q )]
           2  3  3    1  4  4       1       2

(c26) h211:ratcoeff(fc[2],dq[1],2);

(d26) [(l  g  m  - l  g  m ) cos(q ) sin(q )
        2  3  3    1  4  4       1       2

         + (l  g  m  - l  g  m ) sin(q ) cos(q )]
             1  4  4    2  3  3       1       2

(c27) h212:ratcoeff(fc[2],dq[1],1);

(d27)        [0]

(c28) h222:ratcoeff(fc[2],dq[2],2);

(d28)        [0]

(c29) save("fcpbarreduced",fc);

(d29)        [fcpbarreduced, fc]

(c30) closefile();
```

A.2 Conclusion

The Leu and Hemati,[9], macsyma program has been modified to compute the equations of motion for closed link kinematic chain manipulators using the algorithm as presented by Luh and Zheng, [10]. However, the method of extending this program to manipulators that have closed link kinematic chains in combination with open link kinematic chains has not been completely explored. In J. F. Kleinfinger and W. Khahil's paper "Dynamic Modelling of Closed-Loop Robots", [7], they state that they too tried to implement Luh and Zheng's algorithm using a Denavit-Hartenburg convention, but had to develop a new geometric representation convention to make their software work.

Further research into the work of Kleinfinger and Khahil could enhance AFIT's capability to model robotic manipulator configurations greatly.

Appendix B. *MatrixX Programming*

For the numerical integration of the work expressions the following MatrixX program, [16], was used.

This is a typical Vax management system command file used to submit numerical integration jobs in batch mode. It calls up initial conditions needed to start the program and calculates the work for a large section of reachable points in the first quadrant of the workspace. Once the work is determined, the efficiency is then computed.

```
$mws
define 'atan2slp.matx'
execute("icconinvar.matx");
for i=1:36;...
    k(i)=0.02*i;...
  for j=9:20;...
   yf=(j*0.7/20.0);...
   yi=yf+0.1;...
   yaxis(j)=yi;...
   int=20;...
   execute("initialize.matx");...
   execute("integrate.matx");...
    etas(i,j)=abs(mo*a*(yf-yi))/(abs(work1s(i,j))+abs(work2s(i,j)));...
    etap(i,j)=abs(mo*a*(yf-yi))/(abs(work1p(i,j))+abs(work2p(i,j)));...
    etaprime(i,j)=etas(i,j)/etap(i,j);...
j=j+1;...
end;...
i=i+1;...
end;
for i=1:20;...
    k(i)=0.02*i;...
  for j=20:26;...
   yf=(j*0.7/20.0);...
```

```
    yi=yf+0.1;...
    yaxis(j)=yi;...
    int=20;...
    execute("initialize.matx");...
    execute("integrate.matx");...
     etas(i,j)=abs(mo*a*(yf-yi))/(abs(work1s(i,j))+abs(work2s(i,j)));...
     etap(i,j)=abs(mo*a*(yf-yi))/(abs(work1p(i,j))+abs(work2p(i,j)));...
     etaprime(i,j)=etas(i,j)/etap(i,j);...
j=j+1;...
end;...
i=i+1;...
end;
for i=17:36;...
    k(i)=0.02*i;...
  for j=1:9;...
   yf=(j*0.7/20.0);...
   yi=yf+0.1;...
   yaxis(j)=yi;...
   int=20;...
   execute("initialize.matx");...
   execute("integrate.matx");...
    etas(i,j)=abs(mo*a*(yf-yi))/(abs(work1s(i,j))+abs(work2s(i,j)));...
    etap(i,j)=abs(mo*a*(yf-yi))/(abs(work1p(i,j))+abs(work2p(i,j)));...
    etaprime(i,j)=etas(i,j)/etap(i,j);...
j=j+1;...
end;...
i=i+1;...
end;
for i=36:43;...
    k(i)=0.02*i;...
  for j=1:12;...
   yf=(j*0.7/20.0);...
   yi=yf+0.1;...
   yaxis(j)=yi;...
```

```
    int=20;...
    execute("initialize.matx");...
    execute("integrate.matx");...
     etas(i,j)=abs(mo*a*(yf-yi))/(abs(work1s(i,j))+abs(work2s(i,j)));...
     etap(i,j)=abs(mo*a*(yf-yi))/(abs(work1p(i,j))+abs(work2p(i,j)));...
     etaprime(i,j)=etas(i,j)/etap(i,j);...
j=j+1;...
end;...
i=i+1;...
end;
save 'inertws.mdat'
exit
$exit
```

This is the end of the main program. The subroutines called by this command file are atan2slp.matx, icconinvar.matx, initialize.matx, and integrate.matx. These will all be explained as they are presented. The first is atan2slp.matx. This subroutine requires two arguments, x and y position in the workspace, and finds the arctangent resolving any quadrant ambiguity that naturally arises using the not well-behaved inverse tangent function. This is the same as the atan2 function in Fortran.

```
//theta=atan2slp(num,den);
theta=atan(num/den);
if den<0 then;...
  if num>0 then; theta=pi+theta; end;...
  if num<0 then; theta=theta-pi; end;...
end;
if den=0 then;...
  if num>0 then; theta=pi/2; end;...
  if num<0 then; theta=-pi/2; end;...
end;
retf;
```

The next subroutine called is icconinvar.matx. This file initializes a number of constants used in the equations of motion. These consist mainly of link lengths and masses. This file is as follows,

```
a=9.8;
mo=1.6;
g=0;
c=.2;
m1=mo/(12*c);
m2=mo/(6*c);
l1=.36322;
l2=.6705;
mm=.75*mo/c
l1cs=(-mo*l1-m2*l1)/m1;
l2c=(-mo*l2)/m2;
l3=l1;
l4=l2;
m3=m1;
m4=m2;
m2p=m2/4;
l2p=l2/4;
l2pc=(mo*l4-m3*l2)/m2;
l4c=0;
l3c=(mo*l1*l4)/(m3*l2p);
l1cp=(-l1*(mo+m4)-m3*l3c)/m1;
izz1=.000072917*m1;
izz2=.000072917*m2;
izz3=.000072917*m3;
izz4=.000072917*m4;
izz2p=.000072917*m2p;
```

The next subroutine called is initialize.matx. This routine initializes the integration routine by calculating the first quantity of the arrays. Therefore, when integrate takes the difference of the i-1 quantity in the array, it can. This file is as follows,

```
y(1)=yi;
t(1)=0;
work1s(i,j)=0;
work2s(i,j)=0;
```

```
work1p(i,j)=0;
work2p(i,j)=0;
th2(1)=acos(((k(i)**2)+(y(1)**2)-(l1**2+l2**2))/(2*l1*l2));
num=((-(l2*sin(th2(1)))*k(i))+((l1+l2*cos(th2(1)))*y(1)));
den=((l2*sin(th2(1))*y(1))+((l1+l2*cos(th2(1)))*k(i)));
th1(1)=atan2slp(num,den);
    ydot(1)=-a*t(1);
    th1dot(1)=(l2*sin(th1(1)+th2(1))*ydot(1))/(l1*l2*sin(th2(1)));
    th2dot(i)=((-l1*sin(th1(1))-l2*sin(th1(1)+th2(1)))*ydot(1))/...
              (l1*l2*sin(th2(1)));
    yddot=-a;
    th1ddot(1)=(yddot+l2*sin(th1(1)+th2(1))*(th1dot(1)+th2dot(1))**2+...
               (l1*cos(th1(1))*(th1dot(1)**2)+l2*cos(th1(1)+th2(1))*...
               (th1dot(1)+th2dot(1))**2)*(cot(th1(1)+th2(1)))+...
               l1*sin(th1(1))*th1dot(1)**2)/(l1*cos(th1(1))+l2*...
               cos(th1(1)+th2(1))-cot(th1(1)+th2(1))*(l1*sin(th1(1))+...
               l2*sin(th1(1)+th2(1))));
    th2ddot(i)=(l1*sin(th1(1))*th1ddot(1)+l1*cos(th1(1))*th1dot(1)**2+...
               l2*sin(th1(1)+th2(1))*th1ddot(1)+l2*cos(th1(1)+th2(1))*...
               (th1dot(1)+th2dot(1))**2)/(-l2*sin(th1(1)+th2(1)));
    t1s(1)=(m1*l1cs**2+izz1+m1pr*l1prc**2+izz1pr+m2pr*l2prc**2...
           +mm*l2pr**2+izz2pr+mm*l1pr**2...
            +(m2+mm+m2pr)*l1**2+m2*l2c**2+izz2+mo*l1**2+mo*l2**2...
           +2*l1*...
           (-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(1)))*th1ddot(1)...
            +(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+mm*l2pr**2+mo*l2**2...
           +l1*(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(1)))*th2ddot(1)...
           +(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*g*cos(th1(1)+th2(1))...
           +(m1*l1cs+(m2+mm+m2pr)*l1+mo*l1-m1pr*l1prc-mm*l1pr)...
           *g*cos(th1(1));
    t2s(1)=(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+mm*l2pr**2...
           +mo*l2**2+l1*(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(1)))*...
           th1ddot(1)+(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+...
           mo*l2**2+mm*l2pr**2)*th2ddot(1)...
```

```
        +(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*g*cos(th1(1)+th2(1));
t1p(1)=(m1*l1pc**2+izz1+m1ppr*l1pprc**2+izz1ppr+mm*l1ppr**2...
        +m2p*l1**2...
        +m3*l3c**2+izz3+m4*l1**2+mo*l1**2)*th1ddot(1)...
        -(m2p*l1*l2pc+m3*l2p*l3c-m4*l1*l4c-mo*l1*l4)...
        *cos(th2(1))*(th1ddot(1)+...
        th2ddot(1))+(m1*l1pc+m3*l3c+(m4+m2p)*l1+mo*l1-mm*l1ppr...
        -m1ppr*l1pprc)...
        *g*cos(th1(1));
t2p(1)=-(m3*l2p*l3c-m4*l1*l4c-mo*l1*l4+m2p*l1*l2pc...
        )*cos(th2(1))*th1ddot(1)...
        +(m2*l2pc**2+izz2p+m3*l2p**2+mm*l2ppr**2+...
        m4*l4c**2+izz4+mo*l4**2)*(th1ddot(1)...
        +th2ddot(1))-(2*m2p*l2pc+m3*l2p+mm*l2ppr...
        -m4*l4c-mo*l4)*g*cos(th1(1)...
        +th2(1));
```

Finally, the numerical integration is able to be executed. This routine is in integrate.matx and is as follows,

```
for n=2:21;...
    t(n)=(sqrt((2*(abs(yf-yi)))/a))*(n-1)/int;...
    y(n)=yi-(a*(t(n)**2)/2);...
    th2(n)=acos(((k(i)**2)+(y(n)**2)-(l1**2+l2**2))/(2*l1*l2));...
    num=((-(l2*sin(th2(n)))*k(i))+((l1+l2*cos(th2(n)))*y(n)));...
    den=((l2*sin(th2(n))*y(n))+((l1+l2*cos(th2(n)))*k(i)));...
    th1(n)=atan2slp(num,den);...
    ydot(n)=-a*t(n);...
    th1dot(n)=(l2*sin(th1(n)+th2(n))*ydot(n))/(l1*l2*sin(th2(n)));...
    th2dot(n)=((-l1*sin(th1(n))-l2*sin(th1(n)+th2(n)))*ydot(n))/...
              (l1*l2*sin(th2(n)));...
    yddot=-a;...
    th1ddot(n)=(yddot+l2*sin(th1(n)+th2(n))*(th1dot(n)+th2dot(n))**2+...
               (l1*cos(th1(n))*(th1dot(n)**2)+l2*cos(th1(n)+th2(n))*...
               (th1dot(n)+th2dot(n))**2)*(cot(th1(n)+th2(n)))+...
```

```
          l1*sin(th1(n))*th1dot(n)**2)/(l1*cos(th1(n))+l2*...
          cos(th1(n)+th2(n))-cot(th1(n)+th2(n))*(l1*sin(th1(n))+...
          l2*sin(th1(n)+th2(n))));...
th2ddot(n)=(l1*sin(th1(n))*th1ddot(n)+l1*cos(th1(n))*th1dot(n)**2+...
          l2*sin(th1(n)+th2(n))*th1ddot(n)+l2*cos(th1(n)+th2(n))*...
          (th1dot(n)+th2dot(n))**2)/(-l2*sin(th1(n)+th2(n)));...
t1s(n)=(m1*l1cs**2+izz1+m1pr*l1prc**2+izz1pr+m2pr*l2prc**2...
      +mm*l2pr**2+izz2pr+mm*l1pr**2...
       +(m2+mm+m2pr)*l1**2+m2*l2c**2+izz2+mo*l1**2+mo*l2**2...
      +2*l1*...
      (-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(n)))*th1ddot(n)...
       +(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+mm*l2pr**2+mo*l2**2...
      +l1*(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(n)))*th2ddot(n)...
      +(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*g*cos(th1(n)+th2(n))...
      +(m1*l1cs+(m2+mm+m2pr)*l1+mo*l1-m1pr*l1prc-mm*l1pr)...
      *g*cos(th1(n));...
t2s(n)=(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+mm*l2pr**2...
      +mo*l2**2+l1*(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*cos(th2(n)))*...
      th1ddot(n)+(m2*l2c**2+izz2+m2pr*l2prc**2+izz2pr+...
      mo*l2**2+mm*l2pr**2)*th2ddot(n)...
      +(-mm*l2pr-m2pr*l2prc+m2*l2c+mo*l2)*g*cos(th1(n)+th2(n));...
t1p(n)=(m1*l1pc**2+izz1+m1ppr*l1pprc**2+izz1ppr+mm*l1ppr**2...
      +m2p*l1**2...
      +m3*l3c**2+izz3+m4*l1**2+mo*l1**2)*th1ddot(n)...
      -(m2p*l1*l2pc+m3*l2p*l3c-m4*l1*l4c-mo*l1*l4)...
      *cos(th2(n))*(th1dot(n)+...
      th2ddot(n))+(m1*l1pc+m3*l3c+(m4+m2p)*l1+mo*l1-mm*l1ppr...
      -m1ppr*l1pprc)...
      *g*cos(th1(n));...
t2p(n)=-(m3*l2p*l3c-m4*l1*l4c-mo*l1*l4+m2p*l1*l2pc...
      )*cos(th2(n))*th1ddot(n)...
      +(m2*l2pc**2+izz2p+m3*l2p**2+mm*l2ppr**2+...
      m4*l4c**2+izz4+mo*l4**2)*(th1ddot(n)...
      +th2ddot(n))-(2*m2p*l2pc+m3*l2p+mm*l2ppr...
```

```
            -m4*l4c-mo*l4)*g*cos(th1(n)...
            +th2(n));...
    w1s(n)=.5*(t1s(n-1)+t1s(n))*(th1(n)-th1(n-1));...
    w2s(n)=.5*(t2s(n-1)+t2s(n))*(th2(n)-th2(n-1));...
    w1p(n)=.5*(t1p(n-1)+t1p(n))*(th1(n)-th1(n-1));...
    w2p(n)=.5*(t2p(n-1)+t2p(n))*((th1(n)+th2(n))-(th1(n-1)+th2(n-1)));...
    work1s(i,j)=work1s(i,j)+w1s(n);...
    work2s(i,j)=work2s(i,j)+w2s(n);...
    work1p(i,j)=work1p(i,j)+w1p(n);...
    work2p(i,j)=work2p(i,j)+w2p(n);...
    n=n+1;...
  end;...
```

These are the algorithms used to perform the numerical integration of work expression for this analysis.

Bibliography

1. Asada, Haruhiko, Kamal Youcef-Toumi, *Direct Drive Robots: Theory and Practice*, Cambridge, Massachusetts: The MIT Press, 1987.

2. Beer, Ferdinand P., E. Russell Johnston, Jr., *Vector Mechanics for Engineers*, New York: McGraw-Hill Book Company, 1972.

3. Fu, K. S., R. C. Gonzalez, C. S. G. Lee, *ROBOTICS: Control, Sensing, Vision, and Intelligence*, New York: Mcgraw-Hill Book Company, 1987.

4. Hertzberg, H. T. E. "Engineering Anthropology," *Human Engineering Guide To Equipment Design*. Washington D. C.: U. S. Government Printing Office, 1972.

5. Kaplan, Wilfred, *Advanced Mathematics for Engineers*, Philippines: Addison-Wesley Publishing COmpany, Inc., 1981, page 777.

6. Kazerooni, H., S. Kim, "A New Architecture for Direct Drive Robots", *IEEE Journal of Robotics and Automation*, vol. 2 , pages 442-445, June 1988.

7. Kleinfinger, J. F., W. Khahil, "Dynamic Modelling of Closed-Loop Robots", *IEEE Journal of Robotics and Automation*, vol. 2 , pages 401-412, June 1988.

8. Leahy, M. B., Jr., "Dynamics Based Control of Vertically Articulated Manipulators with Variable Payloads", *Accepted for Publication in the International Journal of Robotics Research*, 1988.

9. Leu, M. C., N. Hemati, "Automated Symbolic Derivation of Dynamic Equations of Motion for Robotic Manipulators", *Journal of Dynamic Systems, Mearsurement, and Control*, vol. 108, pages 172-179, September 1986.

10. Luh, J. Y. S., Yuan-Fang Zheng, "Computaion of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms", *IEEE Journal of Robotics and Automation*, vol. RA-1, no. 2, pages 95- 103, June 1985.

11. Meirovitch, Leonard, *Methods of Analytical Dynamics*, New York: McGraw-Hill Book Company, 1970.

12. Petrov, B. A., *Manipulators*, Leningrad: Mashinostroenie Publishing House, 1984.

13. Rivin, E. I., *Mechanical Design Of Robots*, New York: McGraw-Hill Book Company, 1988.

14. Song, Shin-Min, Jong-Kil Lee, "The Mechanical Efficiency and Kinematics of Pantograph Type Manipulators", *IEEE Journal of Robotics and Automation*, vol. 2 , pages 414-420, June 1988.

15. *VAX UNIX MACSYMA Reference Manual*, Cambridge: Symbolics, Inc., 1985.

16. *MATRIXx User's Guide*, version 6.0, Santa Clara: Integrated Systems, Inc., May 1986.

Vita

Steve Parker [redacted] one semester at the University of Michigan - Flint, he enlisted in the Air Force in February 1977. Following basic training,he attended Vietnamese language training at the Defense Language Institue, Monterey, California, graduating in February 1978. From there, Steve attended the Air Force School for Applied Cryptological Sciences and was awarded the Air Force specialty of Voice Processing Specialist in September 1978. Next, he was assigned to the 6922nd Electronic Security Squadron, Clark Air Base, Republic of the Philippines for two years. Upon his returned from overseas he was selected again for language training at the Defense Language Institute, this time for the German language program. During this training, Steve was selected for the Airman Education Commissioning Program, and after graduating from language training was sent to the University of Michigan, Ann Arbor, Michigan from which he received the degree of Bachelor of Science in Aerospace Engineering in December 1983. After being commissioned as a Second Lieutenant from Officer Training School, he was assigned to the Electronic Systems Division, Hanscom Air Force Base, Massachusettes until May 1987. Captain Parker entered the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio in June 1987.

[redacted]

[redacted]

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

AD-A227 618

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFIT/GA/AA/89D-08	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION School of Engineering 6b. OFFICE SYMBOL (If applicable) AFIT/ENY	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Air Force Institute of Technology Wright-Patterson AFB OH 45433-6583	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Configuration Comparison Analysis For The AFIT/AAMRL Anthropomorphic Robotic Manipulator

12. PERSONAL AUTHOR(S)
Steven L. Parker, B.S., Capt, USAF

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
M.S. Thesis	FROM _____ TO _____	1989 December	90

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
12	09		Robotics

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Thesis Advisor: C. H. Spenny
Assistant Professor
Department of Aeronautical & Astronautical Engineering

Approved for public release IAW AFR 190-1
Distribution Unlimited
10 Jan 89

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL C. H. Spenny, Asst. Professor	22b. TELEPHONE (Include Area Code) (513) 255-3517 22c. OFFICE SYMBOL ENY

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

A method of calculating mechanical efficiency was developed as a means of comparing the performance of different types of manipulators. As an initial approach to this problem takes into account inertial and gravitational terms of the robot configurations in addition to a variable payload. The method included developing a numerical integration algorithm to calculate the work done by each manipulator at any point in that manipulator's workspace. The efficiencies of two robotic manipulator configurations that are candidates for the design of the AFIT, AAMRL, Anthropomorphic Robotic Manipulator, (A^3RM), were analyzed. The two designs were a serial open link direct drive manipulator, and the closed parallel kinematic chain direct drive manipulator design by Dr. Asada at M. I. T. The difference between the mainpulators was actual mass and kinematic design.

The efficiency measure used to analyze both manipulators was based on the magnitude of the total work done by the manipulator to move a payload a prescribed distance. The effects of a variable mass payload on efficiency have now been individually examined for the cases when the arm has been "tuned" for some nominal payload by means of compensating for gravity, making the robotic configuration invariant, and decoupling the manipulator's dynamic equations of motion.

An algorithm was developed for calculating the mechanical efficiency for different robotic mainipulator configurations. When the manipulators are gravity compensated for a nominal payload, their efficiency increases dramatically, even when the payload is varied from nominal. In addition, when the configuration is tuned for dynamically decoupling and configuration invariance, efficiency is improved. Finally, for most of the reachable workspace of the manipulators, the parallel manipulator is the most efficient.